AUTONOMOUS AGENTS FOR SERVING COMPLEX INFORMATION NEEDS

BY

PARIKSHIT SONDHI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Prof. ChengXiang Zhai, Chair
Prof. Dan Roth
Prof. Bruce Schatz
Dr. Jimeng Sun, IBM Watson Labs

Abstract

Over the past few decades two prominent paradigms for information seeking in the form of search engines and recommendation systems have been developed. However neither of these is well suited to serve queries representing complex information needs (eg. medical case-based queries). As a result users increasingly turn to web communities such as HealthBoards and Yahoo! Answers making them extremely popular. However, not all queries posted there receive informative answers or are answered in a timely manner.

In this work we present a novel paradigm for information service in which autonomous agents help dissatisfied users in web communities by proactively posting responses to their unresolved queries. The main contribution of this work is to concretely define three application tasks based on this paradigm in the healthcare domain, and show that it is indeed feasible to develop agents capable of generating meaningful responses with a high accuracy.

The first task involved designing an agent for resolving physician case-based queries using literature data. We addressed the problem via methods that utilized available biomedical semantic resources and showed that a precision at 10 of upto 0.48 could be achieved. The second study involved resolving layperson queries on web forums by finding similar discussion threads. This task was more challenging due to noisy nature of forum data and unsuitability of existing semantic resources. We developed novel shallow semantic information extraction techniques for the problem, and our methods utilized them to achieve a best precision at 5 of 0.54. Finally the third task was to design an autonomous agent for resolving general healthcare questions on community question answering (cQA) websites. This task required more detailed semantic information in the form of a database containing precise medical entities, verbose text descriptions, and the relations between

them. These were obtained by using health information websites as an information source. We proposed a principled probabilistic model for the problem, and it was found to resolve over 30% of the questions correctly.

Overall our results clearly suggest that autonomous agents are not only feasible, but can also deliver considerable value to both expert and layperson users of web forums and cQA websites. We believe such autonomous agents have great potential and our work opens up an exciting new area of research.

To my parents Mrs. Geetanjali and Dr. S.M. Sondhi.

Acknowledgments

I wish to express my gratitude to a number of people, who over the course of the past five years, have helped make this dissertation possible. Foremost I wish to thank my advisor Prof. ChengXiang Zhai who has been a great mentor, and a constant source of inspiration. He has helped shape my understanding of research in information retrieval, and has been extremely supportive in letting me pursue my interests. His depth of knowledge and dedication towards students have inspired me both in and outside of work. In him I have found a worthy example to emulate.

I also wish to thank my committee members: Dr. Jimeng Sun, Prof. Dan Roth and Prof. Bruce Schatz. Dr. Sun has collaborated closely on many works discussed in this dissertation. He gave me the opportunity to visit IBM Research and work on cutting edge text analytic applications in healthcare. An experience that has exposed me to the many open challenges in the field. I have also worked closely with Prof. Roth, and learnt a great deal about appropriately formulating and analyzing research problems. He has given several useful suggestions that have helped improve my work. Prof. Schatz, as an expert in healthcare domain, has meticulously helped me plug the gaps in my domain knowledge, so I could precisely identify and convey my research goals and contributions.

My thanks are also due to Dr. Raman Chandrasekar, who gave me the opportunity to work at Microsoft Research and Prof. Julia Hockenmaier and Prof. Kevin Chang, who has been a great teachers. I have had many fruitful discussions with them and they has always been very helpful.

I have also enjoyed greatly, my interactions with students and researchers in the computer science department, particularly in the TIMAN group. It has been a pleasure to participate in various research presentations, tutorials and brain storming sessions that have allowed me to broaden my

horizons and develop new perspectives by learning about other research areas.

No words are sufficient to express my gratitude towards my parents, who have worked hard and made sacrifices all their lives to help me realize my dreams. I hope this dissertation would in some measure, make up for their sacrifices. Without their constant love and encouragement, I would never have succeeded in this endeavor. I'm also thankful to my lovely sister for the many lively conversations that have always kept my spirits high.

Finally I'm indebted to my friends Surbhi Sidana, Riya Singh, Vineet Abhishek and Anika Jain. Surbhi has not only been a great pillar of support, but has also acted as a resident medical expert, helping me understand the nuances of healthcare data. Riya, Vineet and Anika have been no less than a family away from home.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Traditional paradigms of information seeking

Over the past few decades search engines and recommendation systems have become the primary resources for users to seek information. A typical user interaction with a search engine is shown in Figure 1.1. The user submits her information need in the form of a keyword query and then scrolls through a ranked list of retrieved results to find documents that resolve it.

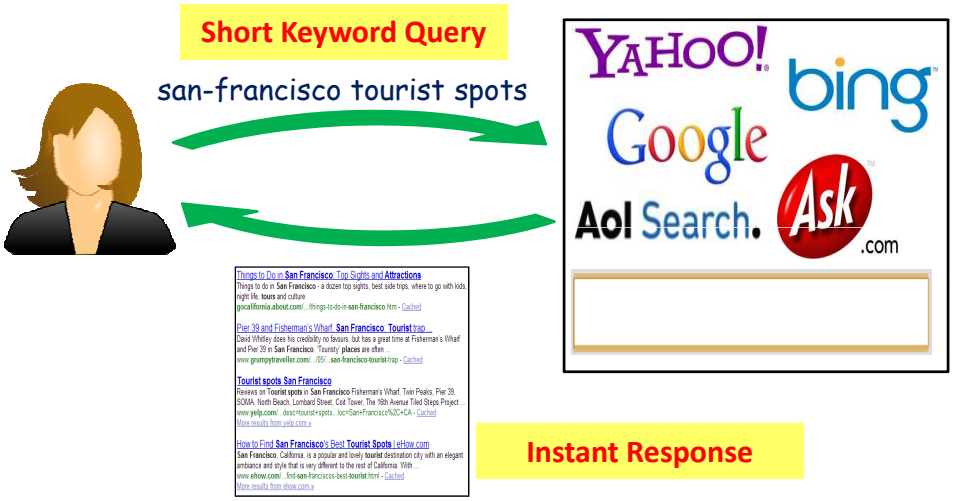

Figure 1.1: Depiction of search paradigm

The recommendation paradigm applies to scenarios where a user's information needs are vague and recurrent. Figure 1.2 shows an example. This approach works well for platforms such as Yahoo! News which recommends news articles of interest to the user, Youtube and Netflix which provide video recommendations and Amazon which provides product recommendations etc.

Figure 1.2: Depiction of recommendation paradigm

Such systems try to model a user's interests by learning from previous examples of objects the user liked. They try to discover some inherent characteristics common to all such objects, and then recommend new objects which possess them as well.

However neither of these paradigms work well in case of complex information needs for which keyword queries are difficult to formulate. Consider an example

What is wrong with my shoulder? I hurt my shoulder on Tuesday diving for a ball and it still hurts a great deal. It has a throbbing pain most of the time and when i push on something or lift my arm up the pain is excruciating. I can't figure out exactly where my shoulder hurts, but rather it feels

like the whole thing hurts. It also seems that for some reason my bicep hurts too. Any ideas on what this could be?

Search engines are good at resolving information needs that translate into short keyword queries (eg. *"San-Francisco Tourist Spots"*).It is unclear how one can formulate a single search query in the above case. A very long query containing all keywords is unlikely to work. On the other hand it is unclear which keywords must be chosen. The only alternate is for the user to formulate several short queries and wade through a large number of partially or completely irrelevant results in the hope of finding the requisite information. This approach requires both a lot of time and skill and does not provide any guarantees towards resolution. Recommendation systems are also naturally unsuitable for such queries as they aren't designed to allow a user to specify precise ad-hoc information needs. As a result users often turn to web-communities for such complex questions.

1.2 Web communities for information seeking

The unsuitability of search and recommendation paradigms for complex information needs implies that users must turn to human experts in web communities such as web forums and community question answering (cQA) services. Liu et. al. [52] show how in many cases failed searches can also lead users to ask questions on web communities and study this transition in detail. As a result web communities have become extremely popular. For example the Yahoo! Answers[1] platform is a premier online question answering service that spans several q-a categories such as health, entertainment, travel, electronics etc. It has over 200 million users, with nearly 15 million visiting daily [78].

However increased popularity does not necessarily imply a high degree of satisfaction. Often questions may go unresolved, poorly resolved or are resolved with a significant delay. Agichtein et. al. [12] showed that upto 50% users on Yahoo! Answers remained dissatisfied by the answers

[1] http://answers.yahoo.com

they received. This thesis proposes a new paradigm of information seeking that attempts to address this issue, i.e., autonomous agents for information service.

1.3 Autonomous agents for information service

In this thesis we introduce a novel paradigm for information service in which an autonomous agent proactively helps dissatisfied users by posting responses to their unresolved questions on web communities. Figure 1.3 presents the details of our idea. The system operates just like another member of the community. It constantly monitors for unresolved queries and whenever possible, generates an appropriate response and posts it. Thus a user can seamlessly get the best of both worlds: human experts and automated systems without having to make any additional effort.

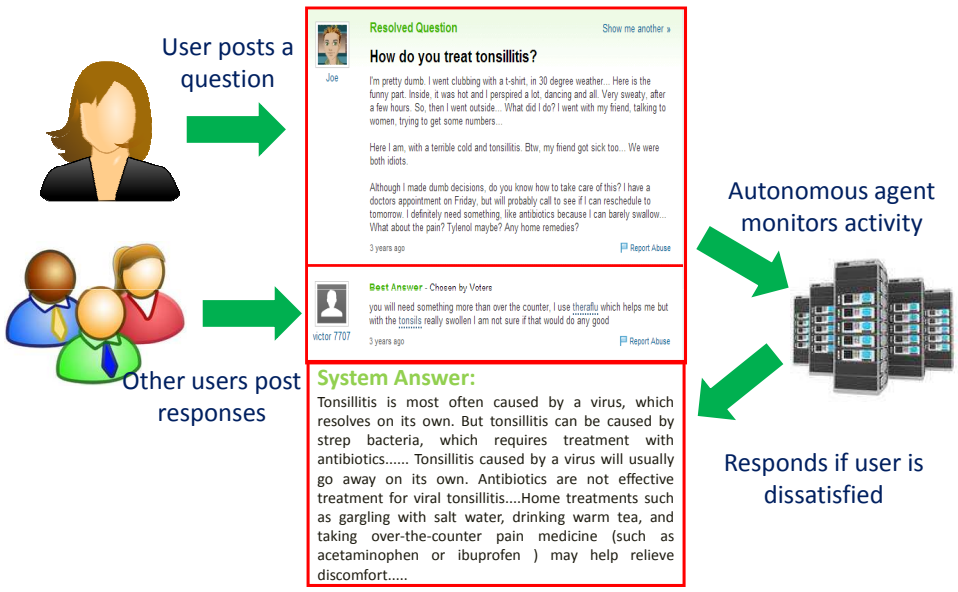

Figure 1.3: Depiction of the novel autonomous agent paradigm

1.3.1 Autonomous agents vs. traditional search

The proposed novel paradigm differs from the traditional search paradigm in several respects. First the information needs that the system is required to resolve tend to be quite complex, more so than those represented by short keyword queries submitted to a search engine. More importantly the query text that system needs to provide a response to, is written by users for human experts and not for an automated system. This makes the task more challenging.

On the other hand an automated agent's asynchronous nature relaxes the strict real time response requirements that a traditional search engine must operate under. This allows for slower but more sophisticated algorithms to be applied for tackling the retrieval challenges.

Another advantage stems from the fact that there is no additional effort required on the part of the user to get the system's results. This makes user expectations and consequently accuracy requirements lower than in a search engine. Consider for example if we tried to deliver the same algorithms for serving complex information needs, through a search style interface with a search box and a real time response. In such a case, the user would need to consciously make the effort of submitting the query into the search interface and then analyzing the retrieved results. However, given that the information needs are quite complex, many of them may be unresolvable leading to the user's time and effort being wasted. In the autonomous agent paradigm, the system can choose to post only those responses whose accuracy it is confident about. Even in cases where a poor response does gets posted, it can simply be ignored by the user just like other poor responses from human experts in the community. As a result user experience does not suffer.

Finally the approach also has an added effect of enriching content. Since more web community pages will now have resolved queries, they will also provide greater utility to other users with similar information needs who may encounter them for example through web search queries.

1.3.2 Autonomous agents for complex health information needs

The primary goal of our thesis is to study the feasibility of such a novel paradigm in the health domain. The major hypothesis to be tested is that it is indeed feasible to construct automated

Task ID	Trageted Web Community	Query Type	Query Source	Response Type	Response Information Source	Semantic Analysis Required
1	Physician forums	Physician medical case-based	Collection of real case-based queries by physicians	Top similar medical cases	Medical Literature	Medical entity identification
2	General health forums	Layperson medical case-based	Queries posted on HealthBoards web forum	Top similar medical cases	Healthcare Web Forums	Shallow information extraction
3	Healthcare cQA websites	General health Questions	Healthcare questions posted on Yahoo! Answers	Text snippet resolving the question	Health Information Websites	Identification of medical entities and their relationships

Table 1.1: Overview of the three application tasks studied in the thesis

agents which are capable of generating meaningful responses to complex healthcare queries with a reasonably high accuracy. To develop such an automated agent, we propose a number of novel approaches that can leverage robust shallow semantic analysis to improve the task performance over the state of the art baseline methods.

We study the problem by defining three application tasks in the healthcare domain. Each task differs in its targeted web community and the information source used (see Table 1.1).

Similar Case Retrieval using Literature Data

The first application task targets web-communities with physicians users (eg. Medscape Connect[2]). Physicians often have medical case-based queries in which they want to find medical cases similar to a given patient case[2]. A sample query is shown below

Query:

Male smoker. The chest X ray shows surgical clips at the right pulmonary apex and a metal endoprosthesis in the right main bronchus. In addition a poorly defined spiculated opacity is seen at the left apex with a diffuse lucency in the left upper hemithorax. On CT images the right lung pulmonary graft appears normal. There is a stellar opacity in the left apex with multiple lung cysts separated by thin walls some confluent with bizarre shapes.

[2]http://www.medscape.com/connect

Our goal is to help physicians resolve such case-based queries by providing automated responses containing research articles discussing similar cases. A sample response is shown below

Intended Response:

Following literature articles may be relevant to your case:

1. *Best Cases from the AFIP Bronchogenic Squamous Cell Carcinoma*
   Source: http://radiographics.rsna.org/content/23/6/1639.full
2. *Malignant Pleural Mesothelioma: Evaluation with CT, MR Imaging, and PET*
   Source: http://radiographics.rsna.org/content/24/1/105.full
3. *T1 Non-Small Cell Lung Cancer: Imaging and Histopathologic Findings and Their Prognostic Implications*
   Source: http://radiographics.rsna.org/content/24/6/1617.full
4. .....

The task is close to a traditional search problem in that we need to retrieve unstructured text documents similar to a given query. However important differences exist. Case-based queries like the one shown above, tend to be quite complex with a lot more keywords than a traditional search query. Thus identifying relative importance of keywords is a challenge. On the other hand additional semantic resources such as medical ontologies (eg. UMLS[9], MeSH[3]) and medical entity extractors (eg. MetaMap[4]) are also available and can help us address it. These resources tend to work well with the well formed technical language of physician queries and literature articles. In chapter 3 we show how semantic information from these resources may be combined with traditional search techniques, and that doing so is sufficient to achieve a reasonably high accuracy for the task.

Similar Case Retrieval using Forum Data

Our next application task targets healthcare web-forums used by laypersons (eg. Healthboards,

Medhelp etc.). Such forums are rife with case-based queries in which laypersons (often patients or their relatives) describe a medical case and want other users in the community to discuss their experiences with similar medical problems. A sample query is shown below

Query:

I am severly allergic to some product that is found in both Tostitos and Doritos, as well as random other types of chips. I know the solution is "don't eat chips" but what could the product be? I don't want to accidentally consume it. When I eat this, I get very bad stomach cramps and it ruins the rest of my day/night - the only solution is to go to sleep so I can't feel it. Help! Any ideas on this?

While the task is applicable to a much larger user base, it comes at a cost of making the problem of generating a useful response harder. Not only do the case-based queries continue to remain complex, they are now also interspersed with a lot of non-case related background information such as *"I don't want to accidentally consume it."*. Any system intending to automatically resolve such queries must first differentiate important case related information from the background. Moreover laypersons are unlikely to understand medical literature. Hence it is unsuited as an information source. We instead propose to use existing forum threads for this purpose, i.e. generate a response redirecting the user to existing threads discussing similar medical cases. A sample response is shown below

Intended Response:

The following threads discuss similar problems:

1. *Doritos Allergy Very Severe and New*
   source: http://www.healthboards...572297-doritos-allergy-very-severe-new.html
2. *Certain Foods + Beer = Flushing and Head Pounding Help!*
   source: http://www.healthboards...head-pounding-help.html

3. *Peanut/Food Allergies*

   source: http://www.healthboards...food-allergies.html

4. .....

Use of forum threads as an information source presents its own unique challenges. Each thread comprises a sequence of posts and cannot as a whole be treated as an unstructured bag of words. Moreover the semantic resources used in the previous task also no longer work well with the noisy and non-technical language found in forum data. In chapter 4 we show that identifying medical entities alone is not sufficient. Instead a high accuracy is achieved via approaches involving sentence level shallow information extraction (see chapter 5) and thread representations with non-uniform post weighing.

Resolving General Health Questions using Health Information Websites

Finally in the third and the most ambitious of the three tasks, we design an autonomous agent for resolving general healthcare questions posted on community question answering (cQA) websites (eg. Yahoo! Answers). The agent uses semi-structured medical content from healthcare websites such as eMedicineHealth[3] and general informational websites like Wikipedia[4] as its information source. Just like medical literature and forum discussion threads, such websites represent a prominent source of medical information. The intended response is a coherent text snippet containing the relevant information along with a reference to the webpage it was obtained from. An example is shown below

Question:

How do you treat tonsillitis?

I'm pretty dumb. I went clubbing with a t-shirt, in 30 degree weather... Here is the funny part. Inside, it was hot and I perspired a lot, dancing and all. Very sweaty, after a few hours. So, then

[3] http://www.emedicinehealth.com

[4] http://www.wikipedia.org

I went outside... What did I do? I went with my friend, talking to women, trying to get some numbers...Here I am, with a terrible cold and tonsillitis. Btw, my friend got sick too... We were both idiots. Although I made dumb decisions, do you know how to take care of this? I have a doctors appointment on Friday, but will probably call to see if I can reschedule to tomorrow. I definitely need something, like antibiotics because I can barely swallow... What about the pain? Tylenol maybe? Any home remedies?

Intended Response:

Tonsillitis is most often caused by a virus, which resolves on its own. But tonsillitis can be caused by strep bacteria, which requires treatment with antibiotics ..... Tonsillitis caused by a virus will usually go away on its own. Antibiotics are not effective treatment for viral tonsillitis .... Home treatments such as gargling with salt water, drinking warm tea, and taking over-the-counter pain medicine (such as acetaminophen or ibuprofen) may help relieve discomfort .....

Source: $http://www.emedicinehealth.com...articlekey = 138098\&pf = 2$

The questions are similar to case-based queries in that they are formulated by laypersons and contain substantial background information. However the expected response is no longer restricted to similar medical cases. The keywords expected in the response can in fact be quite different from those in the question. In the above example, while the question contains phrases such as "cold", "tonsillitis" and "barely swallow" which describe the symptoms, the response must contain phrases like "gargling with salt water" and "drinking warm tea" which discuss treatments. Thus text similarity even when coupled with precise and shallow entity extraction is unlikely to work well. The system needs to not only be aware that "cold" and "tonsillitis" are important medical entities, but also that the expected response must contain text describing their *treatment*. Thus semantic information both in terms of identifiable medical entities and their relationships to text in the information source webpages needs to be incorporated. In chapter 6 we show how this information may be extracted by leveraging the structure commonly found in most healthcare websites and subsequently organized in a relational database. we then derive a principled probabilistic model for generating responses and show that significant performance improvements may be achieved

over traditional retrieval methods. Finally in chapter 7 we also study the problem of automatically ensuring that only reliable health information websites are chosen as resources for the problem.

The three application tasks we study present us with both generic and unique challenges. In each case the final response depends only on the top few most relevant information source documents. Hence all three tasks require a high precision, but are not sensitive to low recall. This characteristic makes the tasks practically viable, provided sufficiently large information sources are available (which is true in all cases). Moreover improved understanding of a complex query through identification of important keywords is a recurring problem across tasks. We resolve this by exploiting different types of semantic information. The semantic analysis performed depends on the type of the query to be resolved.

On the other hand formulating an appropriate representation of the information source is an entirely task specific problem. A forum thread, a literature article and a healthcare website all have their unique characterisitcs and so must be represented quite differently.

The rest of the thesis is organized as follows. Chapter 2 covers a survey of related work. Chapter 3 discusses our work on the first application task i.e. case-based retrieval from medical literature data. Chapter 4 presents the second application task of finding similar cases from web forum data and chapter 5 describes in details the shallow information extraction techniques used for the purpose. Finally chapter 6 covers the third application task of resolving questions on cQA websites and chapter 7 presents some techniques for ensuring the reliability of the response content. Detailed conclusions and directions for future work are covered in chapter 8.

Chapter 2

A Survey of Related Work

2.1 Case-based retrieval using literature data

The problem of information retrieval for biomedical literature has gathered tremendous interest since 2003, when the TREC Genomics Track [7] was introduced. Research has focused on both utilizing general retrieval methods and leveraging semantic resources for the problem. Siadaty et. al. [79] present a high precision retrieval method using sentence-level co-occurrence of query terms to generate relevance scores. Can and Baykal [26] present MedicoPort, a biomedical search engine that uses TF-IDF-based ranking and incorporates semantic knowledge in the form of the UMLS metathesaurus. Lu et. al. [54] found TF-IDF-based methods to be better than sentence-level co-occurrence-based methods using the TREC Genomics track dataset [7]. Huang and Hu [40] present a Bayesian learning method for achieving result diversity in biomedical information retrieval.

Query expansion techniques, including relevance feedback and pseudo-relevance feedback, have been widely used [94, 82, 53]. Lu [53] offers a survey of multiple biomedical search systems that use these feedback techniques. Relevance feedback involves expanding the search query with relevant keywords from the retrieved relevant documents. Pseudo-relevance feedback expands the query with additional keywords from top-ranked documents from an initial search. In both cases, results are re-ranked by searching again with the expanded query. While both methods are generally known to improve performance, the latter is more popular as it does not require labeled relevant documents. Other techniques for query expansion include using ontology (e.g., [18]) or global corpus analysis (e.g. [92]). Previous work on feedback and query expansion has not been tailored for dealing with long and complex case queries, which is what we study in the thesis.

PubMed [6] maintained by the National Library of Medicine is a database of biomedical research articles containing over 21 million citations. PubMed also runs a clinical query search service [5] that allows users to refine search results in specific clinical research areas. Their approach differs from ours in that the refinement is achieved by pre-categorizing the documents in the collection to different clinical research areas via use of study-type filters [89]. They do not attempt to develop case query-specific methods. Another related and more challenging task is biomedical question answering [15], where the goal is to retrieve precise answers to natural language biomedical questions instead of retrieving entire documents as in our task.

In case-based document retrieval, research has focused on developing multi-modal approaches, which utilize both text and image similarity to solve the problem [23, 61, 67, 10, 74, 48]. Text and image similarity scores are calculated using standard methods, and the focus is on developing optimal strategies for combining them. For example, Shao et al. [10] linearly combine text-based and image-based similarity to generate a relevance score. Ruiz [74] also linearly combines the scores from GIFT [1], a standard content-based image retrieval system (GIFT), and SMART [76], a standard TF-IDF-based text retrieval system. Quellec et al. [67] use decision trees to combine image and text similarity features. Our differs from these approaches in that our main goal is to develop new text-based retrieval methods specific for the task.

Our work is closer to work by some other participants in the ImageCLEF 2010 medical case-based retrieval task. Dinh and Tamine [29] extract MeSH concepts appearing in a case query and develop a BM25 [69] based approach for document scoring, with the vocabulary restricted only to MeSH lexicon. Wu et. al. [91] present a TF-IDF-based approach with phrases extracted using MetaMap [4] incorporated into the vocabulary. Our work differs from these approaches in that we do not directly use semantic resources to restrict or extend the vocabulary. Instead we use MeSH thesaurus for pseudo-relevance feedback rather than direct query expansion and MetaMap mappings for semantic keyword weighing. We also develop additional methods based on physician feedback. Overall most of our methods outperform these competing approaches [23].

2.2 Case-based retrieval using forum data

Recent work has explored techniques to improve information access to web forums. Baldwin et al [16] aim at an ILIAD (Improved Linux Information Access by Data Mining) system. Their preliminary work uses classification techniques to estimate the utility of a thread in troubleshooting particular problems. In [27], the authors studied how to extract question-answer pairs from forums.

There have also been numerous efforts to improve the performance of information retrieval tasks by using the inherent structured form of some types of textual data. Duan and Zhai [30] study different thread structure based schemes for smoothing the post language model, to improve forum post retrieval performance. Elsas and Carbonell [32] found that selective methods that only score threads using some posts tend to be more useful for thread retrieval compared to methods that use all posts in the thread. Seo et. al. [77] also show that using correctly annotated thread structures are quite useful in improving retrieval performance. Singh et. al [80] propose a method for estimating thread similarity by decomposing a thread into a set of weighted overlapping components and then calculating lexical similarities between thread components. Finally Elsas et al [31] present retrieval models for blog feed search in which they explore different units of representations of a blog entry as a unit as opposed to the blog feed as a whole. They derive a small document model, which measures how topically related a blog entry is to its feed and this extends well beyond blog feed retrieval, to any document that is a structured collection of smaller units. However none of these methods incorporate semantic information into the relevance ranking function and hence aren't well suited for the long and complex case-based queries encountered in our task.

2.3 Community question answering

Prior work in automatically resolving cQA questions has centered around finding similar questions from within a QA archive. These approaches tend to treat a question thread as a structured document with question and answer fields and focus on identifying the similarity between questions. For example, Jeon et. al. [42] propose a retrieval model which exploits similarity between answers of existing questions to learn translation probabilities, which allows them to match semantically

similar questions despite lexical mismatch. A subsequent work by Xue et. al. [93] combines a translation-based language model for the question field with a query likelihood model for the answer. Other related approaches include finding answers from frequently asked questions on the web [81, 43]. Our work differs from these approaches in that our responses come from health information websites. Thus we may be able to resolve questions that have not previously been asked. Using health information websites as a source also allows us to provide a limited assurance of reliability of information being provided. Since it is possible to automatically predict the reliability of a given healthcare website with reasonable accuracy.

Another related area is that of structured document retrieval, where the focus is on developing retrieval models for documents with well defined fields. Many field based models have been developed in the past including BM25F [70, 49, 50]. The principle idea behind them is that each query keyword may have been intended to match the content of a specific field. Hence they tend to assign different weights to document fields while estimating relevance.

Extensive research has also been done in developing question answering systems. Over the past few years, both open [8] and closed domain systems [15] have been proposed. However most of these approaches are designed to answer only a restricted set of questions such as short and precise factoid [20] or definitional [28]. In addition they tend to require deep natural language processing steps such as generation of parse trees to analyze the syntax and semantics of the question, which are unlikely to work well with noisy cQA data.

2.4 Predicting reliability of health information websites

The quality of medical information on the Web has attracted considerable attention from medical domain researchers. Matthews et al. [59] evaluated a set of 195 webpages pertaining to alternative cancer treatments and found nearly 90% have atleast one flaw. Related studies by Marriott et al. [57] and Tang et al. [84] also concluded that medical information quality on the internet was variable.

The first attempt to automatically identify high quality health information on the Web was pub-

Principle	Description
Authoritativeness	Qualification of the article's authors or reviewers must be present on some webpage on the website.
Complementarity	Information on the webpage should support, not replace, the doctor-patient relationship.
Privacy	Privacy and confidentiality of personal data submitted to the site by the visitor must be respected. This is required only if the page itself requires some personal information to be provided by the user.
Attribution	Source(s) of published information must be cited on some page on the site. This rule is required only if none of the authors or reviewers are qualified medical professionals.
Justifiability	Site must back up claims regarding benefits on some page on the site.
Transparency	Accessible presentation on page and email contact on some page on the site.
Financial disclosure	Funding sources must be identified on some page on the site, if the page is written by a site author.
Advertising policy	Advertising content is clearly distinguished from editorial content.

Table 2.1: HONcode principles for manual website reliability accreditation

lished in 1999, by Price and Hersh [66] who developed a simple rule based system which perfectly separated desirable and undesirable documents using a heuristic scoring function. However their dataset, comprising of only 48 documents, was too small to draw concrete conclusions on either the characteristics of medical webpages or the discriminative power of features. In other related attempts, Aphinyanaphongs and Aliferis [14] used text categorization models for classifying pages discussing unproven treatments, Wang and Richard [87] used a regular expression based heuristic approach for measuring information quality

More recently the Health on Net (HON)[1] foundation has laid out eight widely accepted high level principles that a website must satisfy in order to be considered reliable. These are referred to as HONcode principles and are shown in Table 2.1. However these principles are not directly computable. Hence they are currently only used to guide human experts in manually accrediting healthcare websites. Our goal in this work is to automate this accreditation process. Recently Gaudinat et al. trained classifiers to predict each of the Health on Net reliability criteria (e.g. presence of author names) using content based features [35] and only URL based features [34]. However our approach differs from them in that while we do use features inspired by the eight HONcode principles, we try to predict reliability directly rather than trying to predict individual criteria.

Other related approaches for identifying low quality webpages have focused mainly on detect-

[1] http://www.hon.ch

ing spam webpages through link structures [38, 21], the most popular being Page Rank [24]. Our goal differs from spam detection approaches [17, 13, 98], since we attempt to directly assess reliability of legitimate webpages and analyze the utility of our learnt models in real applications. Other related works include [51, 73, 58].

Chapter 3

Similar Medical Case Retrieval using Literature Data

3.1 Introduction

In this chapter, we study the first application scenario where an agent proactively helps resolve physician case-based queries by finding full-text articles from the literature that discuss similar cases. We address the most important challenge that needs to be dealt with in order to make the agent feasible, namely - development of novel retrieval methods capable of accurately predicting if the medical case discussed in a literature article is similar to the physician's query. We call this task case-based document retrieval.

Beyond being a critical component of our autonomous agent, there are also other important uses for such capabilities, both educationally and in the clinical setting. In today's evidence-based training programs, medical students often find it useful to view information from teaching files [46] containing both detailed case histories and outcomes, as well as test results and imaging studies. This allows individual physicians to enhance their diagnostic skills through self-paced learning. In clinical practice, the capability to review case files and related medical articles from curated collections assembled by prominent clinical institutions may significantly improve physicians' ability to make a diagnosis in complex or puzzling cases [33].

From a technical perspective, case queries are often long unstructured natural language text, containing arbitrary combinations of patient background information, symptoms, test results or diagnosis information, etc. In some cases related images such as x-rays etc. may also be provided. Figure 3.1 shows two examples from the ImageCLEF 2010 case retrieval dataset [23].

These are in contrast with general informational queries in the biomedical domain (e.g. the query: "review article on cholesterol emboli" [39]), which tend to be shorter and more open in

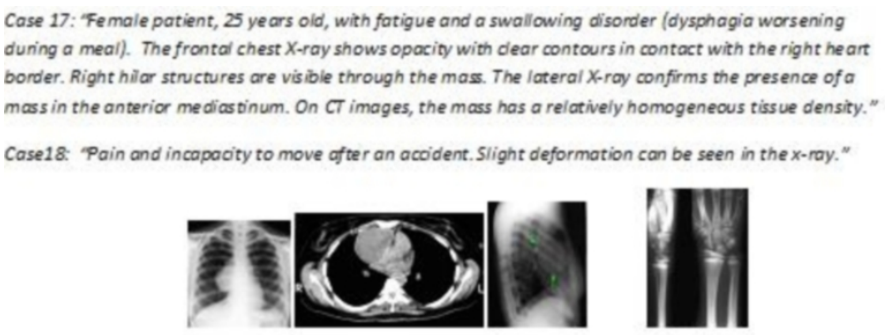

Figure 3.1: Sample case queries and associated images. Case 17 (left three) and Case 18 (right one)

their scope. They also differ from typical queries used in genomics information retrieval where the goal is often to answer a particular question (e.g., finding the molecular function of a gene) [7].

The problem of medical literature retrieval for case queries has gathered interest due to the wide acceptance of evidence-based medicine [90, 36, 75, 68] and interest in the development of medical case-based reasoning systems [11, 19]. To promote further research in the area, the ImageCLEF medical case retrieval task [2] was introduced in 2009 and continued in 2010 and 2011. It contributed to the development of a standardized test collection for the problem. Based on the performance of participating systems, it has been observed that text-based approaches, based on standard document retrieval algorithms, tend to outperform multi-modal methods [23, 61, 63], due to the poor performance of image content-based retrieval. It is thus reasonable to expect that performance could be improved if the standard text retrieval methods were tailored specifically for handling case queries.

Our main contribution in this chapter is to evaluate the utility of general retrieval methods for the task and develop new state-of-the-art text retrieval approaches specific to case queries. Our results show that while well-tuned general retrieval methods work reasonably well, they have several limitations. In particular two stand out:

1. **Vocabulary Gap:** A user may use a different vocabulary in the query than that used by authors of relevant documents to describe the same concept. Alternatively the concepts expected in response may not be frequent in the query.

2. **Non-Optimal Query Term Weighing:** Failing to differentiate the important query keywords from the less important ones

These challenges may be overcome through the use of also affords us the opportunity to address them by exploiting additional language resources such as UMLS and MeSH tags assigned to the documents.

To address these challenges, we propose to extend a general retrieval method by performing query term reweighting based on UMLS thesaurus [9] and pseudo-relevance feedback based on MeSH thesaurus [3]; both were found to improve retrieval accuracy. Additional information from physicians in the form of related query keywords was also found to be helpful, while relevance feedback improved performance moderately. Our system based on these strategies achieved the best performance in the ImageCLEF medical case retrieval challenge 2010 [2].

3.2 Dataset description

The ImageCLEF dataset comprises 5585 research articles from Radiological Society of North America Radiographics journal (http://radiographics.rsna.org/) and a total of 19 case queries with relevance judgments provided. Five of the 19 queries are from the 2009 task and 14 from the 2010 task. The case queries were formulated based on cases from the teaching file Casimage [72]. This teaching file contains cases (including images) from radiological practices that are used by clinicians mainly for teaching purposes. Any information regarding diagnoses and treatments was removed from these queries, so as to simulate the situation of the clinician who has to diagnose the patient. For the judging process, however, the relevance judges (experts in the field) were provided with complete information in order to perform accurate evaluation. We used the five queries from the 2009 task for parameter tuning and the remaining 14 queries from 2010 for evaluation. The evaluation set contained on average 37.2 relevant documents per query (min=2, max=97, std. dev =31.01).

3.3 Design objectives

Since much of the task setup in case retrieval is similar to general retrieval, we hypothesize that the state-of-the-art general retrieval models may achieve reasonable performance. On the other hand, the challenges arising as a consequence of long complex queries must be addressed by using semantic resources and getting additional information from the users. Based on this intuition, the main objectives of our experiments were to test the following hypotheses:

- **H1:** State-of-the-art general retrieval methods will achieve reasonable performance for the case-based document retrieval task.
- **H2:** Performance can be improved by systematically addressing limitations of the state-of-the-art methods, via use of medical thesauri.
- **H3:** Performance can be improved via user feedback, in the form of additional query keywords and relevance judgments.

We began by analyzing the performance of one of the best performing general retrieval methods: KL-divergence Retrieval Model with Dirichlet Smoothing and pseudo-relevance feedback [95] to test H1. To the best of our knowledge, it has not been applied to the case-based document retrieval task before. Then using it as a baseline, we developed additional methods to test H2 and H3.

3.4 Method

In this section we start with a discussion of the standard retrieval model that forms our baseline. We then discuss our proposed task-specific approaches based on thesauri and physician feedback. Our strategy will be to use the baseline search method as a black box (for searching full-text articles) and implement additional methods on top of the baseline method, by either appropriate expansion and weighing of queries or re-ranking of results.

3.4.1 Standard retrieval models

KL-divergence retrieval model with dirichlet smoothing

Language modeling [95] provides a sound statistical framework for designing retrieval models. One of the best-performing retrieval models based on language modeling is the Kullback-Leibler (KL) divergence retrieval model [95]. Given a query Q and a document D, this model would first estimate a unigram query language model Q (i.e., a word distribution) based on a given query and a document language model θ_D for document D, and then score the document D with respect to query Q based on negative KL-divergence between the two language models, $-D(\theta_Q||\theta_D)$, defined below:

$$-D(\theta_Q||\theta_D) = -\sum_{w \in V} p(w|\theta_W) log \frac{p(w|\theta_D}{p(w|\theta_Q)}$$

where V is the set of words in our vocabulary, and $p(w|\theta_Q)$ and $p(w|\theta_D)$ are the probabilities of word w given by the two language models, respectively. The negative KL-divergence intuitively measures the similarity of the query language model and the document language model. Thus it would favor a document that matches more query words. The document language model θ_D characterized as the word distribution $p(w|\theta_D)$ is usually estimated using Dirichlet prior smoothing [95]:

$$p(w|\theta_D) = \frac{c(w, D) + \mu p(w|C)}{|D| + \mu}$$

where $c(w, D)$ is the count of word w in document D, $p(w|C)$ is a background/reference language model estimated based on all the documents in the collection and helps providing probabilities for words unseen in a document, and μ is a smoothing parameter, which was tuned using training data. The optimal value was found to be $\mu = 4800$. The simplest way to estimate the query model θ_Q is to set $p(w|\theta_Q)$ to the relative frequency of a word in the query:

$$p(w|\theta_Q) = \frac{c(w,Q)}{\sum_{w \in Q} c(w,Q)}$$

Since this approach assigns zero probability to words not in the query, a potentially better way to estimate this model is to use a technique called pseudo-relevance feedback, which we discuss next.

Pseudo-Relevance feedback

The basic idea of pseudo-relevance feedback is to treat a small number of top-ranked documents in the initial retrieval result as if they were relevant documents and extract useful terms from these feedback documents to improve the estimate of a query language model. In our experiments, we used the mixture model approach described in [64], which is one of the best-performing approaches to pseudo-relevance feedback. This method first obtains a word distribution characterizing the topic in the feedback documents. Then, it interpolates that word distribution from the feedback documents with the word distribution estimated based on relative frequency of words in the query. The mixture model pseudo-relevance feedback method has a few parameters, which were tuned using the five queries from ImageCLEF 2009 dataset. The best results were found when using only the top two documents for feedback.

3.4.2 Thesaurus-Based approaches

We propose two approaches for exploiting available thesauri in the medical domain to improve the general retrieval model.

Semantic query weighing

Not all case query keywords are equally useful in identifying relevant documents. General retrieval uses Inverse Document Frequency (IDF) [56] as a critical heuristic for weighing keywords, which assumes that it is more important to match a rare term than a frequent term. However, this general

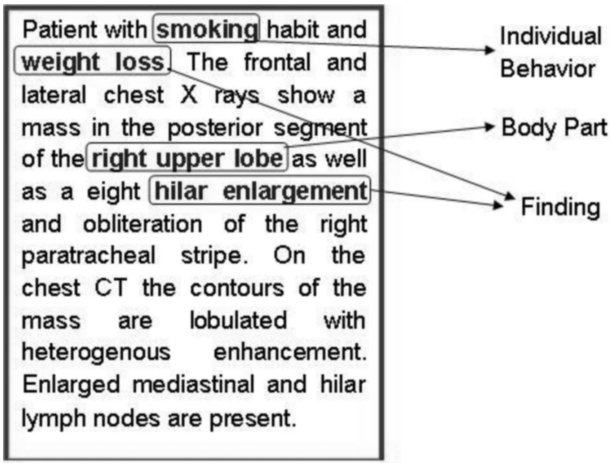

Figure 3.2: A sample query with semantic types of some important keywords identified

heuristic is insufficient for our task since keywords belonging to certain semantic categories like *disease names, symptoms, drugs* etc. (see Figure 3.2) are more representative of a medical case and must be assigned high weights regardless of their IDF.

To this end, we propose to map all query keywords to UMLS [9] semantic types using the MMTx toolkit [4], and assign weights according to their semantic types. Based on our analysis of all the semantic types appearing in the training queries, the following were among the most discriminative for retrieving a relevant case:
Disease or Syndrome, Body Part organ or organ component, Sign or Symptom, Finding, Acquired Abnormality, Congenital Abnormality, Mental or Behavioral Dysfunction, Neoplasm, Pharmacologic Substance

To highlight their importance, the weights of query keywords mapped to these types were doubled i.e. their query count was modified as $c'(w, Q) = c(w, Q) * 2$.

MeSH-based pseudo-relevance feedback

The primary motivation behind any case query is often to find *potential diagnoses* for it. In other words, an ideal case retrieval system could be thought of as executing the following steps:

1. Make a list of *potential diagnoses* for the case query at hand.

2. Assign a high relevance score to all documents discussing these diagnoses.

Assuming the case query is sufficiently descriptive, there would only be a small number of potential diagnoses possible. If we can predict these conditions and increase the rank of the documents that primarily talk about them, we should be able to improve performance. This breaks down into two problems:

1. How to find out which conditions a given document talks about?

Each medical literature article indexed in PubMed is manually assigned a set of indexing terms from the MeSH thesaurus [3]. It is possible to filter out condition/disease related MeSH terms to identify the prominent conditions the document talks about.

2. How to find out what conditions the query case is likely to represent?

This is a harder problem. In the following discussion, we present two ways of dealing with it.

Top-N-based MeSH feedback

This approach is similar to pseudo-relevance feedback. We make a list of all condition-related MeSH terms present in the top $N = 10$ documents in the initial ranked list generated by the baseline method. We then slightly reduce the weight of any documents below these top N that do not share any MeSH terms with this list (See Figure 3.3).

The approach does have limitations in that it cannot re-rank the top N documents and may not perform well if none of the top N documents are relevant. We overcome this limitation in our second approach.

Distribution-based MeSH feedback

This method is based on the intuition that a MeSH term assigned to a document that contains a large number of query keywords is more likely to represent the query. Thus for each MeSH term, we first identify all the documents indexed with it, and then count the number of unique query

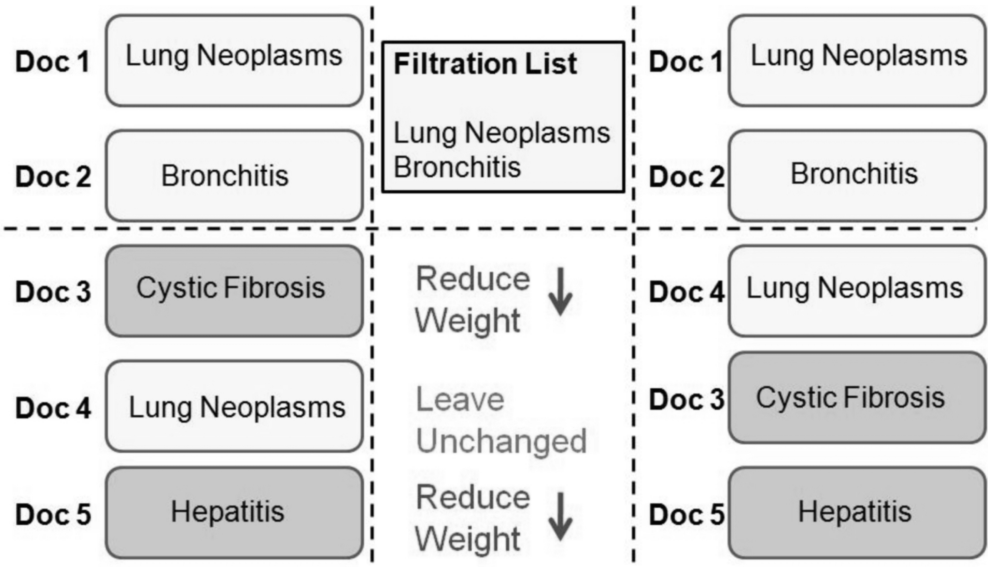

Figure 3.3: Top N based pseudo-relevance feedback using MeSH thesaurus with $N = 2$ documents. Keywords inside the boxes are MeSH terms for the corresponding documents. The left column indicates the order of the original retrieval results. The right column indicates the order after MeSH feedback reweighting.

keywords present in these documents. Finally we pick the 25 highest scoring MeSH terms for our *Filtration List*. This approach is useful in that it can allow us to re-rank the top results also. This becomes important when a high precision is required. A detailed description of the algorithm is given below:

Let M be the set of all condition-related MeSH terms. Then for a given query Q, the method works as shown in Figure 3.4:

Note that for both MeSH-based approaches, we do not take into account the hierarchical relationships between concepts. Incorporating the hierarchy will introduce new parameters which may lead to overfitting on our small training set.

3.4.3 Physician feedback

In an interactive retrieval system, a physician can potentially provide feedback in two ways: (1) additional keywords can be provided to better focus the query; (2) relevance judgments on retrieval results can be provided as examples for the system to use for better inference of relevance. We

1. For every MeSH term m in M, set Score(m) =0
2. Retrieve a ranked list of all the documents L for the query Q
3. For each document d in L
    a. Identify the set of query keywords Sd (subset of Q) found in d
    b. Identify the set of MeSH terms Md (subset of M) found in d
    c. For each MeSH term m in Md:
        i. For each query keyword q in Sd:
            1. If we have never encountered the keyword qbefore in a document labeled with MeSH term m, then Score(m) = Score(m) +1
4. Sort all MeSH terms m in M in descending order of Score(m).
5. Select the top N=25 MeSH terms from the ranked list
6. Re-rank documents by reducing the weights of all documents not labeled with any of these N selected MeSH terms.

Figure 3.4: Distribution based MeSH feedback

now discuss how we can leverage these two different forms of feedback.

Additional query keywords

While submitting a case query, physician users often have a number of additional keywords in mind that they feel are relevant to the case. But they avoid using these keywords as they may influence the results too strongly. However, if used appropriately, these keywords can be very useful in retrieving the right documents. To examine this hypothesis, we asked two physicians to look at each case query along with its associated images (without looking at any relevant documents) and provide additional keywords they thought were relevant. These were then merged to form the additional physician keywords for each query. An example is shown in Figure 3.5.

Original Query:

Female patient, 25 years old, with fatigue and a swallowing disorder (dysphagia worsening during a meal). The frontal chest X-ray shows opacity with clear contours in contact with the right heart border. Right hilar structures are visible through the mass. The lateral X-ray confirms the presence of a mass in the anterior mediastinum. On CT images, the mass has a relatively homogeneous tissue density.

Additional Physician Keywords:

Thymoma, Lymphoma, Dysphagia, Esophageal obstruction, Myasthenia gravis, Fatiguability, Ptosis

Figure 3.5: Additional keywords provided by physicians

We added them to the original query with comparatively low weights to keep them from dominating the results (i.e. their query count was modified as $c'(w, Q) = c(w, Q) * 0.3$). This helped greatly in improving performance. Adding these keywords directly to the query with equal weights did not do as well.

Relevance feedback

To leverage feedback information in an interactive retrieval system, we also experimented with physician relevance feedback. The idea was to ask a physician to judge 20 top-ranked documents as relevant or non-relevant and then use these relevant documents for relevance feedback, i.e., the system would learn from these examples to improve retrieval performance. More specifically, our system can use the same mixture model that we had used for pseudo-relevance feedback to improve the estimation of query language model based on judged relevant documents

3.5 Experiment design

3.5.1 Experimental design

Our first set of experiments was to evaluate the performance of the baseline retrieval method using the standard implementation provided in the LEMUR retrieval toolkit (www.lemurproject.org). This gave an estimate on the best performance that general retrieval methods can achieve for this task. Subsequently, we experimented by adding our proposed methods. Figure 3.6 provides a summary of all the different experiments. The light colored boxes show experiments in which performance was improved by adding a new method on top of all parents. The dark colored boxes show experiments where the performance dropped. More details are available in Table 3.2.

3.5.2 Evaluation criteria

Performance of each method is measured using Precision at 10 ($P@10$), Recall at 30 ($R@30$) and Mean Average Precision (MAP). $P@10$ represents the percentage of relevant documents in the top 10 results. $R@30$ represents the percentage of all the relevant documents in our collection

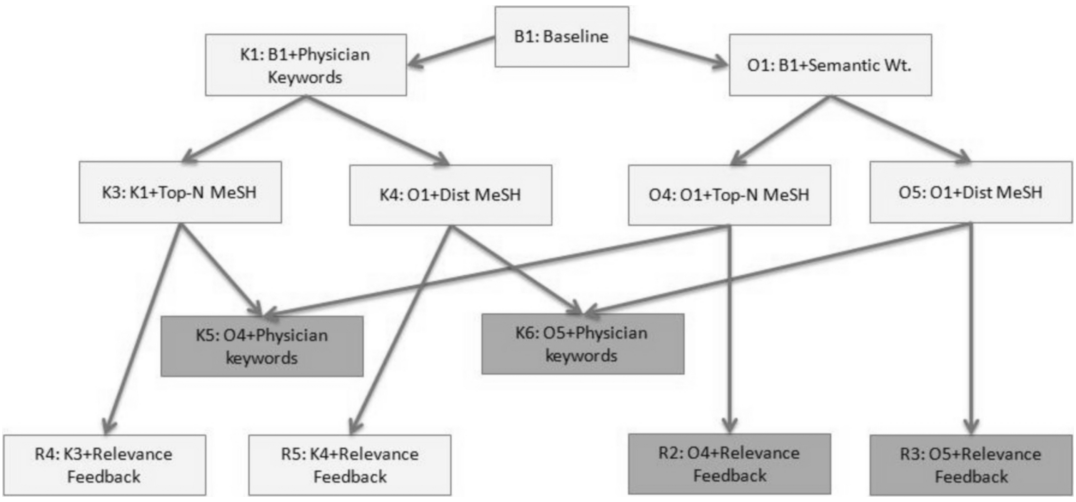

Figure 3.6: Dependency of different runs. Light colored boxes are the runs that improve MAP score over both their parent runs. Dark colored boxes are the runs that reduce MAP score from at least 1 parent run.

that are present in the top 30 results. Mean Average Precision is the arithmetic mean of average precision values over a set of queries. Suppose, for some ranking $r_i \in R$, there are k_i relevant documents in the whole collection. Further, let $rank(j)$ be the rank of j^{th} relevant document and $P(rank(j))$ be the precision at the rank of the j^{th} relevant document. Then

$$P(rank(j)) = \frac{\#RelevantDocumentstillrank(j)}{\#Documentstillrank(j)} = \frac{j}{rank(j)}$$

Average precision of some ranking $r_i \in R$ and the mean average precision over a set of rankings R , are then given by:

$$AP(r_i) = \frac{\sum_{j=1}^{k_i} P(rank(j))}{k_i}, MAP(R) = \frac{\sum_{r_i \in R} AP(r_i)}{|R|}$$

AP intuitively captures the average of precision at every point when a new relevant document is retrieved.

Query ID	Rel. Docs	Precision@10	Recall@30	Average Precision
1	31	0.3	0.2903	0.1931
2	2	0	0	0.0075
3	13	0.5	0.4615	0.3414
4	42	0.2	0.0714	0.0501
5	23	0.6	0.5217	0.4455
6	6	0.2	0.3333	0.0995
7	12	0.7	1.0000	0.6344
8	97	0.3	0.1237	0.3048
9	59	0.7	0.1864	0.178
10	93	0.6	0.1290	0.3403
11	59	0.6	0.2203	0.3366
12	4	0.1	0.7500	0.1304
13	45	0.3	0.0889	0.1657
14	35	0.9	0.5714	0.6284
Avg Over all 14	37.2(31.01)	0.4286(0.26)	0.3392(0.29)	0.2754(0.2)

Table 3.1: Baseline Performance of individual queries. $+/-$ values in () indicate the standard deviations.

3.6 Results

3.6.1 Standard retrieval method

The query-specific performance results are shown in Table 3.1. Since users are more likely to formulate a new query rather than search for relevant documents beyond the third page or so, $R@30$ helps give a good estimate on practical recall. The $P@10$ varies from 0 to 0.9, with a mean of 0.4286 (std. dev. = 0.26). Thus, on average, the baseline shows about 4 relevant documents in the top 10. This suggests that our first hypothesis of baseline performance being reasonable holds. We did however notice a fairly high variation in the performance values. We present a detailed failure analysis of the baseline and some of our other approaches in the discussion section.

3.6.2 Thesaurus-Based methods

Performance comparison of the semantic weighing and MeSH based methods with the baseline is shown in Table 3.2. We observe that semantic weighing (Run O1) generally improves performance across all measures. The improvement in recall is the most pronounced. Of the two MeSH-based approaches, the Top-N approach (Run O2) with $N = 10$ and penalty factor set at 0.1 (tuned using ImageCLEF 2009's 5 queries) improves MAP by 2.5%. The $P@10$ does not change, as the top 10 documents are not re-ranked in this case. On combining with the semantic weighing method

(Run O4), we get the best thesaurus-based run in terms of MAP with an improvement of 6.8%. The Distribution-based MeSH feedback approach (Run O3) performs worse than the baseline. It however starts to show improvement when combined with other methods (Runs O5, K4, K6, R5). This approach also re-ranks the top 10 documents, and when combining it with semantic weighing (Run O5), we observe the best performance in terms of $P@10$ among all the thesaurus runs.

3.6.3 Physician feedback

Additional query keywords

The results in Table 3.2 show that using additional keywords from physicians (Run K1) generally improved performance. The performance improved for 9 cases and was hurt in 3. The magnitude of improvement was higher (40.1%) compared to the previous methods. It thus suggests that a better way to submit a case query is to not only provide the natural language case description, but also any additional ad hoc keywords that the user feels may be moderately relevant. These keywords are often helpful in sufficiently focusing a query, especially when physicians provide potential diagnosis keywords like *lung cancer* which tend to be highly discriminative. Combining methods also resulted in an improvement in performance. This is expected, as the methods address different baseline limitations. Also the MeSH based methods are likely to perform better as the performance of the underlying system improves. The only exceptions are runs that combine semantic weighting with physician keywords (Runs K2, K5, K6). In these cases the performance is generally better than their counterparts without physician keywords (Runs O1, O4, O5), but worse than those without semantic weighing (Runs P1, K3, K4). Overall the best improvement was 42.4% (Run K3) in terms of MAP, 11.7% in terms of $P@10$ and 37.7% in terms of $R@30$.

Relevance feedback

For the relevance feedback experiments, a physician user was shown the top 20 documents for each query. The user was asked to judge the presented documents as relevant or non-relevant. The subsequent (unseen) documents were then re-ranked using the relevance judgments provided

		Performance			% Improvement Over Baseline		
Run ID	Run Name	MAP	$P@10$	$R@30$	MAP	$P@10$	$R@30$
B1	Baseline	0.2754	0.4286	0.3392	–	–	–
Thesaurus Based Runs							
O1	Sem. Wt.	0.2808	0.4429	0.3407	2%	3.3%	0.4%
O2	Top-N MeSH $(10, 0.1)$	0.2824	0.4286	0.3383	2.5%‡	–	−0.2%
O3	Dist. MeSH $(40, 0.1)$	0.2699	0.4214	0.3082	−2.0%	−1.7%	−9.1%
O4	Sem. Wt.+Top-N	0.2942	0.4429	0.3679	6.8%	3.3%	8.5%
O5	Sem. Wt. + Dist. MeSH	0.2908	0.4500	0.3483	5.6%	5%	2.7%
Additional Keyword Runs							
K1	Phy. Keys	0.3858	0.4643	0.4124	40.1%#	8.3%	21.5%
K2	O1 +Phy. Keys	0.3441	0.4714	0.4288	24.9%†	10%	26.4%
K3	O2 + Phy. Keys	0.3922	0.4643	0.3967	42.4%‡	8.3%	17%
K4	O3 + Phy. Keys	0.3897	0.4786	0.4004	41.5%‡	11.7%	18%
K5	O4 + Phy. Keys	0.3599	0.4714	0.4670	30.7%#	10%	37.7%
K6	O5 + Phy. Keys	0.3521	0.4571	0.4392	27.9%#	6.7%	29.5%
Relevance Feedback Runs							
R1	Rel. Feedback ($N = 20$)	0.2840	0.4286	0.3401	3.1%	–	0.3%
R2	O4 + Rel. Fb.	0.2875	0.4429	0.3577	4.4%	3.3%	5.5%
R3	O5+Rel. Fb.	0.2878	0.4500	0.3467	4.5%	5%	2.2%
R4	K3+Rel. Fb.	0.3972	0.4643	0.4171	44.2%*	8.3%	23%
R5	K4+Rel. Fb.	0.3980	0.4786	0.4241	44.5%*	11.7%	25%

Table 3.2: Combination results. All improvements are over the baseline run B1. Statistically significant improvements in MAP (via Wilcoxon signed rank test [88]) are highlighted with superscripts.
$*$ Significant using Wilcoxon signed rank test at level $p < 0.01$
‡Significant using Wilcoxon signed rank test at level $p < 0.025$
Significant using Wilcoxon signed rank test at level $p < 0.05$
†Significant using Wilcoxon signed rank test at level $p < 0.1$

by the user. The $P@10$ of all relevance feedback runs remained the same as that of the original ranking since the top 20 results were not re-ranked. While relevance feedback is known to improve performance [43], in our case it helped moderately in some cases and not at all in others. Runs R1 through R5 in Table 2 show the results of incorporating relevance feedback to some of the major thesaurus-based and additional keyword-based runs. Run R1 which merely adds relevance feedback on top of baseline achieves a 3.1% improvement. On the other hand, runs R2 and R3 show that adding relevance feedback on top of the best performing thesaurus runs (O4 and O5) degrades performance. In case of the best performing additional keyword runs (K3 and K4) we again observe a small improvement (R4 and R5). Overall we achieve the best performance in terms of all the different measures when applying relevance feedback over a combination of additional keywords and distribution based MeSH feedback. Performance of the Top-N based approach is also similar.

Measure to Optimize	Physician Keywords Available	Physician Keywords Unavailable
Precision@10	Baseline + Phy. + Dist.MeSH (K4)	Baseline + Sem. Wt. + Dist. MeSH(O5)
Recall@30	Baseline+Sem. Wt.+Phy.+Top-N MeSH(K5)	Baseline + Sem. Wt. (O1)
MAP	Baseline + Phy. + Top-N MeSH(K3)	Baseline + Sem. Wt. +Top-N MeSH(O4)

Table 3.3: Best method configurations based on application settings.

3.6.4 Algorithm recommendation

Table 3.3 presents the most suitable method configuration depending upon application settings. It also suggests the possibility of constructing a system that dynamically picks up the most suitable configuration. For example if the user supplies additional keywords and indicates that $P@10$ needs to be optimized, the system can generate a ranking using configuration K4. Such a hybrid system would presumably have a higher overall utility for users than any of the individual methods.

3.7 Discussion

In this chapter we presented a study of methods for retrieving medical literature articles for case queries. Our results show that a $P@10$ of over 0.4 was achieved in case of all methods and 0.48 for the best performing method. This figure is quite reasonable (five relevant documents in top 10) and suggests that it is indeed feasible to automatically find similar cases from the literature, and post them in response to queries. Moreover we observed that the incorporation of biomedical semantic resources was found to be clearly beneficial in improving performance.

While our results are encouraging, there are still some challenges that need to be dealt with before the methods can be deployed in practice. We discuss them below.

Small dataset size

The main limitation of our work arises from small size of the evaluation set. A realistic system in a clinical setting will need to deal with hundreds of thousands to millions of documents. Creation of such a large labeled training collection requires time and expertise. Even though the ImageCLEF 2010 dataset is realistic in that the queries are based on real user needs, at 19 queries and 5585 documents, it is fairly small and does not capture the wide spectrum of documents and queries

expected in real life. Currently we deal with the issue by performing statistical significance tests (Wilcoxon signed rank test [88]) to ensure the performance improvements are significant.

Failure analysis

We analyzed individual queries with poor retrieval accuracy (as reflected by low precision) to understand why our methods did not work well with some of them. In general, it seems that poorly performing queries were often ambiguous. In particular two kinds of failures stood out.

Difficulty in recovering new treatments and rare alternate diagnoses

Such documents usually have low keyword overlaps with the query. Our MeSH based and semantic weighing methods are tuned towards retrieving documents with similar diseases and hence do not do as well in such scenarios. The problem is compounded by the fact that, in cases where the disease itself is not the focus of the document, relevant disease MeSH terms may not have been assigned, making the MeSH-based pseudo-relevance feedback methods less effective. Physician keywords help greatly when they overlap with such documents, but this does not always happen.

Confusion between similar diseases/conditions

For example in a query related to *arm fractures*, some *neck fracture* related documents were also retrieved, due to high keyword overlap. In general it was difficult to automatically guess when certain diseases must be logically ruled out.

The two failure modes are related in the sense that, while the first requires methods to have a high recall, the second requires higher precision. The first failure can be potentially alleviated by diversifying the retrieval results so that documents with a rare diagnosis would have a better chance to be ranked on the top, while the second problem can be addressed by using more specific units than single words (e.g., phrases) for indexing. Ultimately, however, overcoming these challenges will require us to incorporate more complex concept models for relationships between biomedical concepts. For example, we can use the hierarchical relationships between concepts within MeSH for more refined feedback.

Physician keywords

Among the many different methods we tried, the incorporation of appropriately weighted physician keywords proved to be the most useful in improving performance. Our physicians were able to provide them fairly quickly, without looking at any relevant documents. This suggests that the technique would be relatively easy to apply in a clinical setting. While these keywords were provided by experts, a natural future task is to generate them automatically. Our methods based on extracting useful MeSH terms for a query are a step in this direction. However MeSH is a controlled vocabulary, while physician keywords are often ad hoc. As a result, for our queries, we found only a limited correspondence between them. One possibility may be to use the co-occurrence of MeSH descriptors in document collections to suggest related keywords.

Chapter 4

Similar Medical Case Retrieval Using Forum Data

4.1 Introduction

In this chapter, we study the second application scenario where an agent proactively helps resolve medical case-based queries found on web forums by finding other discussion threads that discuss similar cases. Healthcare forums such as HealthBoards[1] provide a platform to users for getting answers to their medical case queries. A typical forum thread contains a case query in its first post, and a discussion around it in subsequent posts. An example is shown in Figure 4.1.

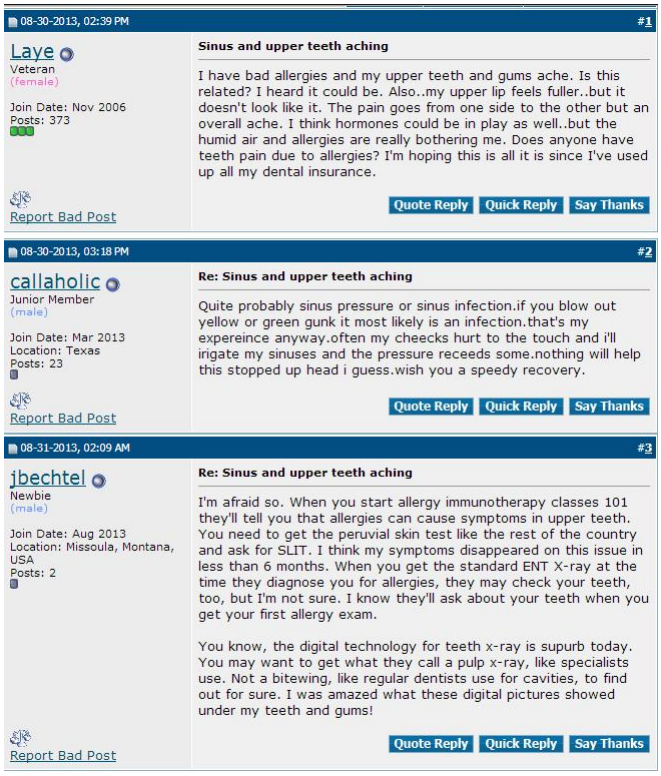

Figure 4.1: Sample healthcare discussion thread

[1] http://www.healthboards.com

However prior research has shown that a sizeable percentage of users' queries aren't satisfactorily resolved [12]. One way to help these users is by exploiting an existing collection of discussion threads. Often many users suffer from the same medical condition and start multiple discussion threads on very similar case queries. As a result many queries may be resolved by directing users to relevant previously existing threads in the collection. Due to its free availability and simple language, forum threads can be an effective source of information for laypersons. Our goal in this chapter is to study whether it is feasible to develop sufficiently accurate case-based forum retrieval methods, that can be used by an autonomous agent to generate useful responses.

From a high level computational point of view, we would treat the first post of an unresolved thread as a medical case query, and retrieve similar threads from a collection. This setup differs from the literature retrieval task in two important respects. First the case query, which is formulated primarily to elicit responses from other laypersons in the community can contain a fair amount of background non-medical information such as emotional statements like *"Help me!"* or *"I'm fed up with this"* etc. It therefore becomes necessary to separate medical case related information from such background. This task becomes especially difficult since medical entity extractors don't always work so well due to the noisy nature of forum data. The second difference stems from the nature of documents to be retrieved, i.e. relevant forum threads. A forum thread is more than just a bag or sequence of words. It is subdivided in a sequence of well defined posts and this internal structure must be considered when formulating a representation.

In this chapter we explore a plethora of methods for the task. In particular we discuss how sentence level shallow information extraction methods may be utilized to improve performance. We also present novel post weighing techniques which utilize a thread's well defined internal structure to improve retrieval effectiveness. We evaluate the methods on a collection of over 350K healthcare forum threads and report promising results.

4.2 Formalizing the forum case retrieval problem

We treat the problem of similar thread finding as a specialized retrieval task with first post of an unresolved thread as a query and the each thread in our existing thread archive as a document. In this section we start by providing some definitions to make the problem more precise and then discuss the various design objectives which guide the development of our methods.

Definition 1 (Forum Post): A forum post p is a sequence of words in a vocabulary set V.

Definition 2 (Forum Thread): A forum thread t is a sequence of posts, i.e., $t = [p_1, ..., p_L]$, where p_i is the i-th post in the thread. In subsequent discussion we will also frequently refer to a forum thread as a document.

Definition 3 (Collection): A collection C is defined as a set of forum threads $C = \{t_1, t_2, ...t_n\}$, where t_i is a thread.

Definition 4 (Case Query): A case query q is defined as the sequence of words in a vocabulary set V which is input to the system for finding similar cases.

Our goal in forum case retrieval is to, given a query q assign a relevance score $Score(q, t_i)$ to each thread t_i in the collection C and return a list of top 5 threads ranked based on their relevance scores as output.

4.3 Methods for forum case retrieval

Our high level approach to solving the problem is similar to the one we employed in the previous chapter for literature retrieval. We start by evaluating the performance of a state of the art baseline retrieval method and then extend it by incorporating various task related characteristics to improve performance. We loosely categorize our methods into 4 different categories.

1. **Baseline:** Inovolve a direct application of state of the art retrieval techniques to the problem.

2. **Semantic Weighing:** Involve extracting and assigning a higher weightage to medical case related keywords in the case query.

3. **Post Weighing:** Involve designing novel keyword weighing techniques that don't treat the content of all posts in a thread equally.

4. **Combination Methods:** Involve combining suitable semantic and post weighing techniques.

In subsequent sections we discuss these methods in detail.

4.3.1 Baseline approaches

We use the popularly used $BM-25$ retrieval model as a baseline for the task. $BM-25$ was originally developed by Robertson et. al. [71] for TREC ad-hoc filtering task and has since been extremely popular as a state of the art general retrieval method. The function can be efficiently computed even on fairly large collections. The relevance score of a thread t to a query q is computed as

$$Score(q,t) = \sum_{w \in V} log \frac{N - n(w) + 0.5}{n(w) + 0.5} \frac{c(w,t)(k_1+1)}{c(w,t) + k_1(1 - b + b\frac{b|t|}{avgtl})} \frac{(k_3+1)c(w,q)}{k_3 + c(w,q)} \tag{4.1}$$

where $w \in V$ represents a word in vocabulary V. N is the total number of threads in the collection C. $n(w)$ represents the number of threads in the collection that contain word w. $c(w,t)$ and $c(w,q)$ represent the frequency of w in the thread t and query q respectively. Finally $|t|$ is the length of a thread in terms of total count of all words appearing in it and $avgtl$ is the average length

of all the threads present in the collection. The value of parameters k_1, k_3 and b are generally set between $1-2$, $0-1000$ and 0.75 respectively.

We use two different baseline approaches based on how the content of a thread is represented.

Thread $BM-25$ ($TBM-25$)

Under this method a thread is considered as a bag of words containing all its posts. We give equal importance to each post and the thread keyword frequency $c(w,t)$ of each word w is calculated by counting all its occurrence in all the posts.

First post $BM-25$ ($FPBM-25$)

Under this method the thread keyword frequencies are obtained by considering only the keywords in the first post of the thread. Content of all subsequent posts are ignored. This approach assumes that the first post in a thread is likely the most representative of its case and hence its keywords are most critical.

4.3.2 Semantic weighing approaches

The goal of semantic weighing approaches is to help identify query keywords that are most representative of the case, and weigh them separately from the background. This is achieved by perturbing the query frequency $c(w,q)$ of certain keywords by introducing additional parameters. We introduce two methods that operate at different levels of granularity

Medical entity extraction

Our first semantic weighing approach was to use an out of the box medical entity extractor to identify individual medical case related entities from the query text. Various medical entity extractors are available for the purpose, but only ADEPT[55] has been specifically trained on medical forums. The algorithm is based on Conditional Random Fields, and the authors have shown that it achieved $F1$ score of 0.84 while all the other algorithms that were trained on non-medical forum

domains,including MetaMap[4] which we used for literature data, achieved $F1$ scores of below 0.5.

In the baseline method, the count of a word in a query $c(w, q)$ is given as the number of occurrences of w in q. In this method, once we have applied a medical entity extractor, all word occurrences are either labeled as being a medical entity or not. Let $\#med(w, q)$ be the number of occurrences of word w that are labeled as a medical entity in query q and $\#nonmed(w, q)$ be the number that aren't. We then replace the count $c(w, q)$ in equation 4.1 by an altered count with a tunable parameter α_m.

$$c'(w, q) = \alpha_m * \#med(w, q) + \#nonmed(w, q)$$

While the approach is easy to apply and can identify entities with a fairly high precision, it doesn't necessarily work so well on forum text, where many keywords representing non-standard medical entities may also be very important. For example in the query shown in Figure 4.2 we find that even though terms like *Tostitos* and *Doritos* are crucial to representing the patient's case, they are not identified as medical entities. This issue is dealt with in our next approach, where we perform sentence level extraction.

Shallow medical information extraction

The second approach operates at a coarser level of granularity. In this case we label entire sentences as being representative of medical information, rather than individual keywords. Each sentence in the query text is assigned one of the following three categories. An example of this kind of labeling is shown in Figure 4.3

1. Physical Examination (PE): The sentence contains the description of diseases, symptoms etc.
2. Medication (MED): The sentence provides description of treatment, medications or other measures taken to resolve the disease.

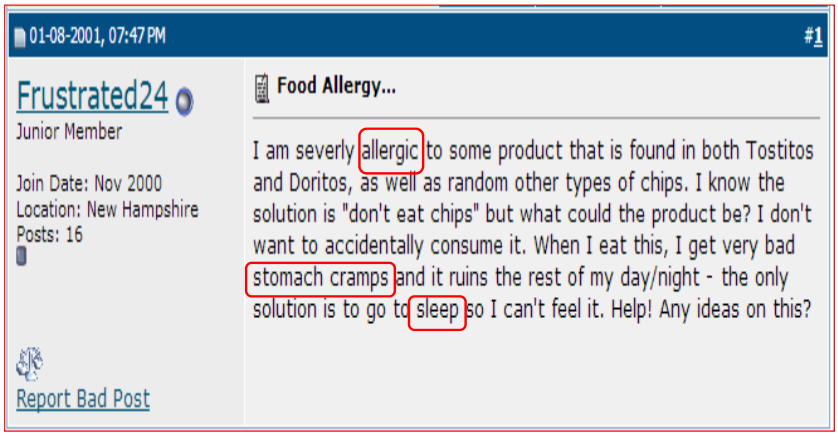

Figure 4.2: Sample medical entity extraction using ADEPT. *allergic, stomach cramps* and *sleep* are identified as medical entities. Tostitos and Doritos are not identified.

3. Background (BKG): Sentence is not covered in either PE or MED. Often covers sentences exhibiting emotional response.

The main intuition behind such a labeling is to separate the critical case related sentences from the background. We restrict ourselves to only three classes, since it is possible to train reasonably accurate classifiers for them and at the same time they are sufficient to represent the most prominent types of sentences appearing in medical forum texts.

The extraction is performed using a Support Vector Machine [22] based classifier, which we found to be most suitable for the task. For a detailed discussion on the definition, methods and analysis of the shallow extraction problem refer to the next chapter.

Once the labeling of sentences is complete, we employ a weighing technique similar to that in the previous section. Only this time two parameters are introduced. Let $\#pe(w,q)$, $\#med(w,q)$ and $\#bkg(w,q)$ be the number of time word w appears in PE, MED and BKG labeled sentences of query q. The modified relevance scoring function is obtained by replacing the query count $c(w,q)$

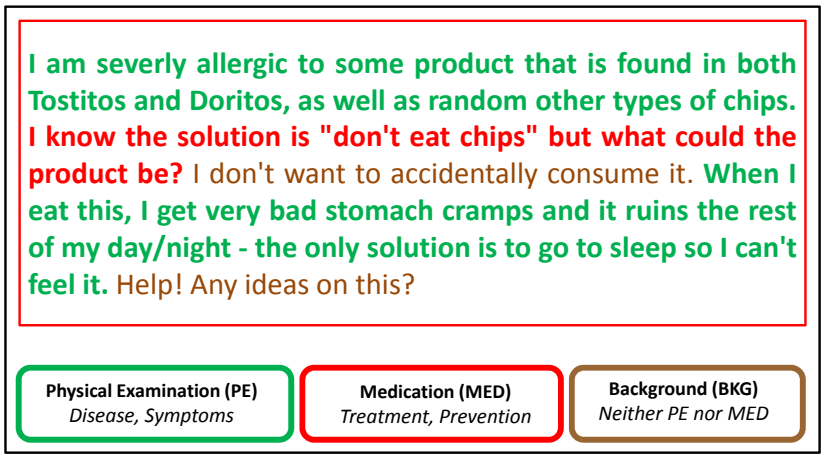

Figure 4.3: An example of **PE**(green),**MED**(red) and **BKG**(brown) sentences.

in equation 4.1 by $c'(w,q)$ defined as

$$c'(w,q) = \alpha_{pe}\#pe(w,q) + \alpha_{med}\#med(w,q) + \#bkg(w,q)$$

where α_{pe} and α_{med} are tunable parameters.

4.3.3 Position based post weighing

So far, we have always treated each post in a thread equally. However, intuitively, not all posts in a thread are equally good in reflecting the problem being discussed in a thread. Indeed, the first post of a thread often defines the medical case to be discussed. The following posts possibly suggest solutions to the problem posed or veer into other lines of discussion. As a result, not all posts are equally representative of the thread and non-uniform weighting of posts is presumably beneficial.

The challenge then is to assign non-uniform weights to posts so that we can potentially improve the content representation of a thread. Below we propose two different schemes: Monotonic Post

Weighing and Parabolic Post Weighing.

In both schemes, weights are assigned to posts based on their relative position in the thread. Specifically, let t be a thread with K posts $[p_1, p_2, ...p_K]$. The weight $pw(p_i, t)$ of a post p_i in t is then solely a function of the position variable i and the number of posts in the thread K i.e.

$$pw(p_i, t) = f(i, K), 1 \leq i \leq K$$

Once defined, the weight is incorporated in the relevance scoring function by replacing the thread word count term $c(w, t)$ in equation 4.1, by an altered thread term count $c'(w, t)$ defined as

$$c'(w, t) = \sum_{i=1}^{K} f(i, K)c(w, p_i) \tag{4.2}$$

where $c(w, p_i)$ is the count of word w in post p_i. See Figure 4.4 as an example. The exact definition of the function $f()$ varies in the two schemes. We use $f_m()$ to represent monotonic and $f_p()$ to represent parabolic post weighing functions.

Monotonic post weighing

In this scheme, we hypothesize that the representativeness or the importance of a post in a thread reduces as its position in the thread increases, i.e., the later a person posts in a thread, the lesser is the weight of the post. This is best represented by a weighting scheme that monotonically decreases the weight assigned to posts as their positions increase. In our experiments, we use the following function:

$$f_m(i, K) = \frac{(\frac{1}{i})^{\beta_m}}{\sum_{j=1}^{K}(\frac{1}{j})^{\beta_m}} \tag{4.3}$$

where β_m is a decay parameter that can be tuned to adjust the rate at which the weight of the post reduces as the post's position increases. The weight normalization included in the denominator ensures that the weights assigned to posts of a thread sum up to unity. When $\beta_m = 0$, this transforms

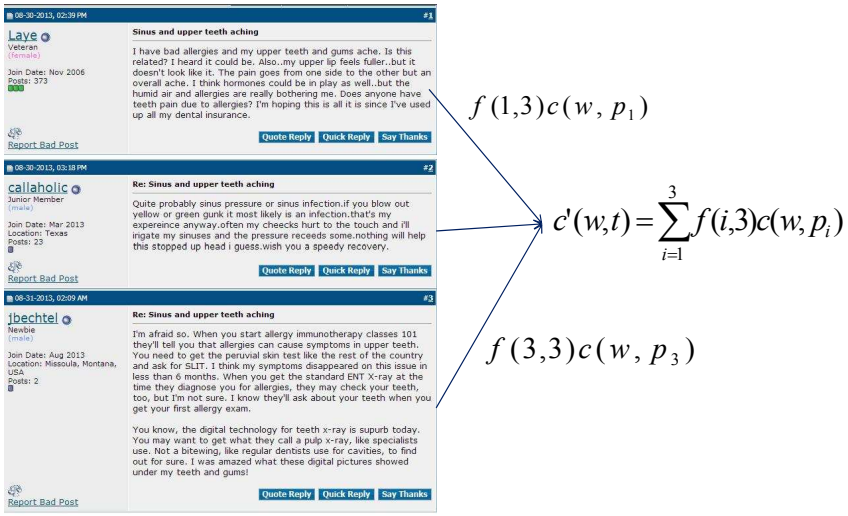

Figure 4.4: Sample post weighing for $K = 3$. $f(i, K)$ gives the weight of post i in a thread with K posts.

to all posts getting equal weights. As β_m increases, the drop in post weights is more pronounced and less gradual. An example is shown in Figure 4.5.

Parabolic weighting of posts

The parabolic weighing scheme is based on the observation that, oftentimes, discussion in a thread stops once a solution to the problem has been posted. Hence intuitively while the initial posts tend to represent the problem well, the posts towards the end are more likely to be representative of the solution. Since the critical aspects of a thread are likely to depend both on the problem keywords as well as the solution keywords, we should assign higher weights to both the initial and the final posts of a thread. This is well modeled by the following skewed parabolic function:

$$g_p(i, K) = \frac{(i - \beta_p K)^2}{(1 - \beta_p K)^2} f_p(i, K) = \frac{g_p(i, K)}{\sum_{j=1}^{K} g_p(j, K)} \tag{4.4}$$

where K is the total number of posts in the thread and $\beta_p \cdot K$ is the position at which the post weight is minimum. We first calculate a function $g_p(i, K)$ which is parabolic w.r.t the position

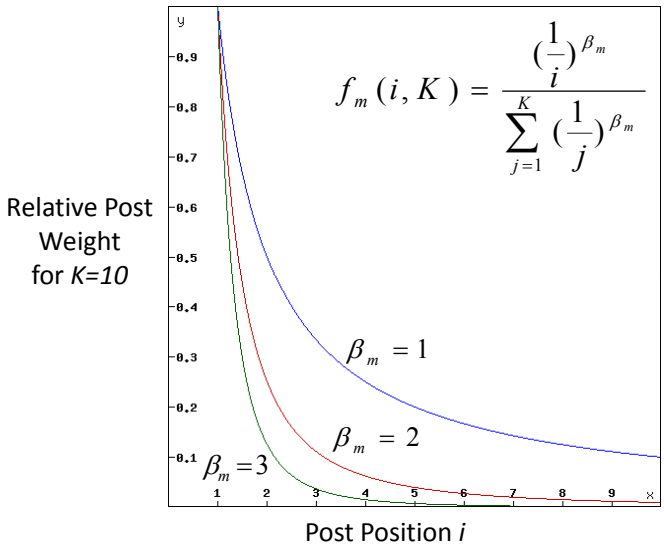

Figure 4.5: Monotonic post weight decay curves for various values of β_m and $K = 10$

variable i and obtain the final post weights $f_p(i, K)$ by normalizing to ensure a unity sum for the weights. This normalization also eliminates the divide by zero exception when $\beta_p = \frac{1}{K}$.

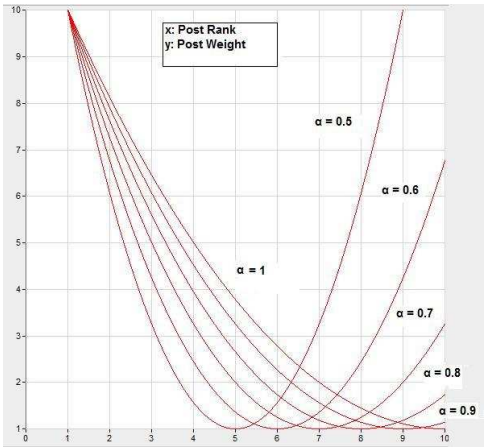

Figure 4.6: Graph depicting variation of post weights w.r.t. β_p in a thread with $K = 10$ posts

The variation of post weights with respect to β_p is as shown in Figure 4.6. As can be seen, $\beta_p = 0.5$ represents the case where the initial posts of a thread are weighted equally with the final posts. As β_p increases, the initial posts are assigned higher and higher weights relative to the final

posts. When $\beta_p = 1$, the weighting function transforms into a monotonically decaying function with the final posts assigned the least weight. Suitable values of β_p are generally found to be around 0.7 [62].

4.3.4 Combination methods

From the method details discussed above it is clear that while semantic weighing techniques alter query word counts, the post weighing techniques alter thread word counts. Thus it is also possible to combine these methods together. More specifically $c'(w, q)$ is generated using one of the semantic weighing methods and $c'(w, t)$ is generated using one of the post weighing methods. These are then plugged into equation 4.1 for generating relevance scores. Since each technique represents a different heuristic, we expect the performance of combination methods to be better than their constituent methods.

4.4 Experiments

4.4.1 Evaluation set construction

Our document collection comprised $350K$ threads crawled from HealthBoards[2], which is the largest healthcare forum on the web. The evaluation was done using 20 queries. Judgments were created via pooling [47], a strategy commonly used in information retrieval evaluation. For each query, top 10 retrieved threads from all our methods were pooled together and judged as either relevant or non-relevant by a human expert. In all over 730 query-thread pairs were judged by two judges. Of these 324 threads were found to be relevant and 406 irrelevant.

In order to ensure consistency in judgments, the two judges first both labeled the same set of 100 query-thread pairs to check for inter-annotator agreement. The only annotation guideline was to consider the similarity between the symptoms and intent of the query and the retrieved thread while making the judgment. 88% of the judgments were found to be in agreement and Cohen kappa

[2] http://healthboards.com

Method	$P@5$	$R@30$	MAP
Baseline			
$TBM-25$	$0.3000(-36.2\%)$	$0.2846(-42.8\%)$	$0.1977(-40.4\%)$
$FPBM-25$	0.4700	0.4975	0.3316
Semantic Weighing			
$FPBM-25+MedEx$	$0.4600(-2.1\%)$	$0.4283(-13.9\%)$	$0.2918(-12.0\%)$
$FPBM-25+ShallowEx$	$0.5300(12.7\%)$	$0.4847(-2.5\%)$	$0.3481(4.9\%)$
Post Weighing			
$MonotonicBM-25$	$0.5100(8.5\%)$	$0.5240(5.3\%)$	$0.3631(9.5\%)$
$ParabolicBM-25$	$0.5100(8.5\%)$	$0.5040(1.3\%)$	$0.3494(5.4\%)$

Table 4.1: Results for methods when applied individually on top of baseline. Values in () are performance improvements over $FPBM-25$

was found to be 0.76. This suggested a reasonably high agreement implying that the annotation task was fairly well defined.

4.4.2 Experiment design

Our first goal was to look into evaluating the performance of our proposed methods, both individually and in combination. We were especially interested in understanding which method combinations are most suitable under differing application requirements, represented by the various evaluation metrics. Methods were evaluated using 5-fold cross validation. The criteria were the same as those discussed in the previous chapter (see section 3.5.2).

The second goal was to look into the sensitivity and stability of parameters. We have introduced a number of parameters through our various methods. We wanted to study study how sensitive the performance is to some of the important parameters. Such an analysis can provide us with valuable insights on the stability of our proposed techniques.

4.5 Results

4.5.1 Individual performance analysis

Table 4.1 shows the results of different methods when applied independently. Among the baseline methods we observe that when a thread is represented using only its first post ($FPBM-25$) the performance is significantly higher than when it is represented using all posts ($TBM-25$).

However, when keywords in subsequent posts are included through monotonic or parabolic post weighing, an 8.5% improvement over $FPBM - 25$ is observed. This is because the content, and hence the keywords, of subsequent posts are only partially related to the medical case being discussed in the thread. Hence subsequent posts must contribute less towards the thread counts of words.

Among the semantic weighing methods, we observe that keyword weighing based only on medical entity extraction ($FPBM - 25 + MedEx$) performs slightly worse than the baseline. On the other hand the sentence level shallow extraction method ($FPBM - 25 + ShallowEx$) performs the best among all individual methods in terms of precision. This is because sentence level extraction also allows us to capture the importance of non-medical keywords which, in the non-technical language of forums, are useful in representing a medical case.

4.5.2 Combined performance analysis

Method	$P@5$	$R@30$	MAP
$FPBM - 25$	0.4700	0.4975	0.3316
$FPBM - 25 + ShallowEx$	$0.5300(12.7\%)$	$0.4847(-2.5\%)$	$0.3481(4.9\%)$
$MonotonicBM - 25 + ShallowEx$	$0.5400(14.9\%)$	$0.5354(7.6\%)$	$0.3745(12.9\%)$
$ParabolicBM - 25 + ShallowEx$	$0.5100(8.5\%)$	$0.5155(3.6\%)$	$0.3573(7.8\%)$

Table 4.2: Results for multiple method combinations. Values in () are performance improvements over $FPBM - 25$

We next look at the performance of combining shallow extraction with different post weighing techniques. Results are shown in Table 4.2. We observe that monotonic post weighing when combined with shallow information extraction, achieves the best performance in terms of all three metrics. Thus clearly the two techniques are compatible. Two queries and their sample similar threads found using the method are shown in Figure 4.7. On the other hand, while parabolic does improve recall and MAP, it performs worse (0.5100) in terms of precision than shallow extraction alone (0.5300).

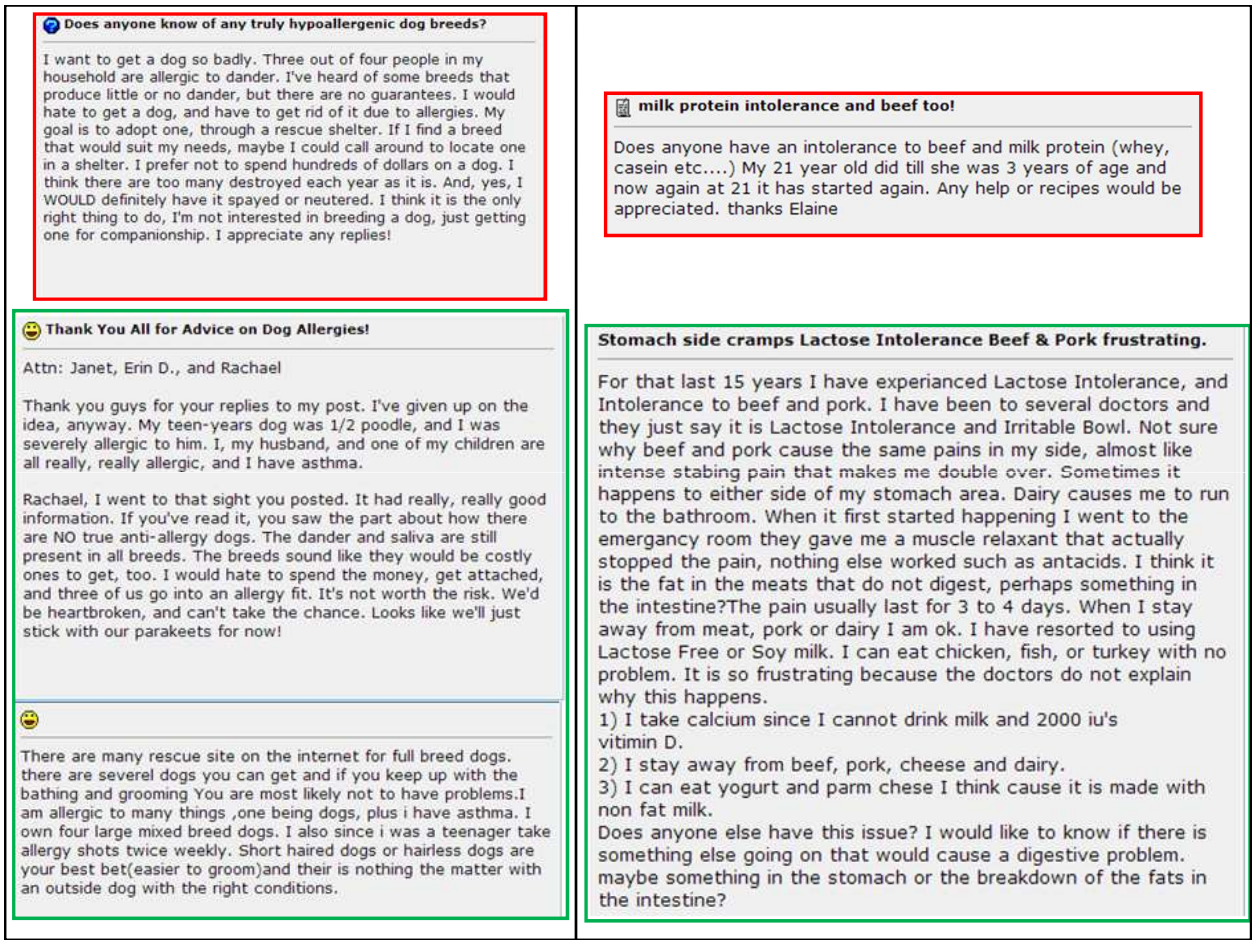

Figure 4.7: Two case-based queries (in red boxes) and their found similar threads (green boxes)

4.5.3 Parameter analysis

Finally we look at analyzing the sensitivity of the two post weighing parameters β_m and β_p. For these experiments, we vary the value of the parameter being analyzed and evaluate its performance on all 20 queries. The results are shown in Figures 4.8 and 4.9. We observe that when varying the two parameters in the expected range of values, the performance varies gradually rather than fluctuating. This suggests that the parameters are easy to set and minor perturbations in their values are unlikely to significantly hurt performance.

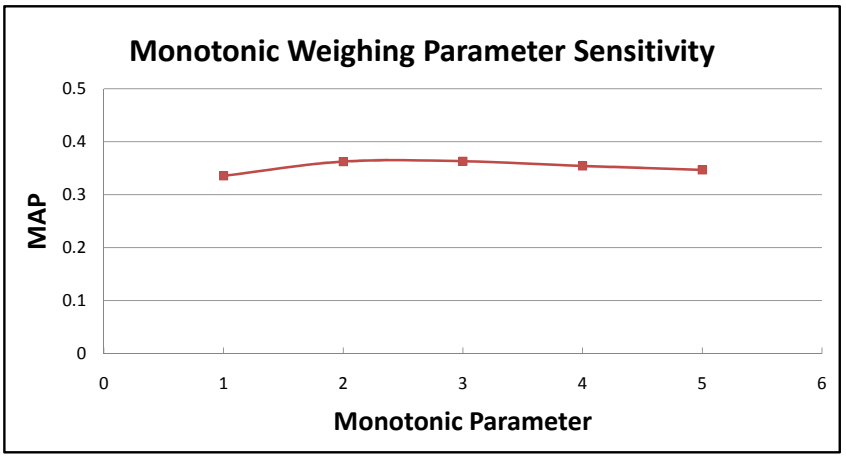

Figure 4.8: Performance variation for parameter β_m on all 20 queries

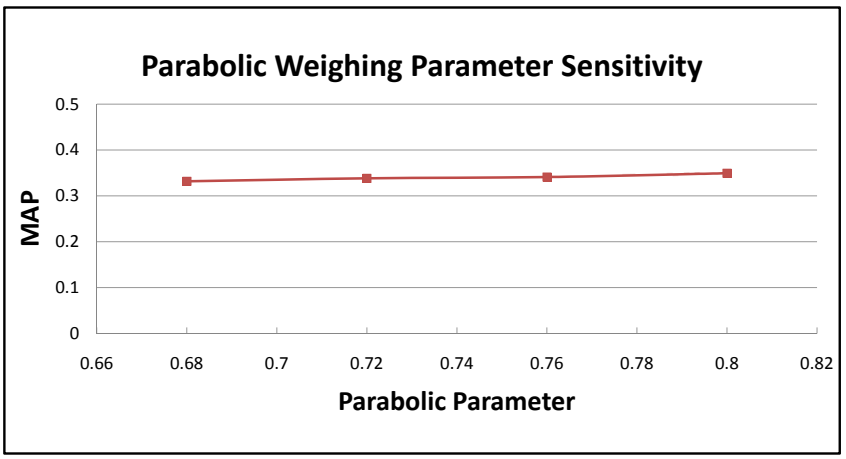

Figure 4.9: Performance variation for parameter β_p on all 20 queries

4.6 Discussion

In this chapter we presented our work on an autonomous agent for healthcare forums. It resolves medical case-based queries present in the first post of unresolved threads, by generating a response containing top 5 threads that discuss medical cases most similar to the unresolved case. We explored a number of different approaches for the problem that attempted to extend the state of the art general retrieval methods for the task, by incorporating semantic and thread structure based information.

The observations suggested that the task is clearly feasible. It does however require a different kind of semantic information than the literature retrieval problem discussed in the previous chapter. Forum queries are formulated mainly by laypersons and hence contain non-technical language and a lot of non-case related background information. As a result even a state of the art medical entity extractor is unable to identify all of the important case related keywords. We instead need sentence level extraction of important case related sentences to separate them from background sentences. For our work three classes PE, MED and BKG seemed sufficient for the task. The resulting method when combined with rank based monotonic post weighing scheme achieved the best performance. We present a detailed discussion of shallow semantic information extraction techniques in the next chapter.

Chapter 5

Shallow Information Extraction over Forum Data

5.1 Introduction

In this chapter we present our work on development of shallow information extraction techniques which are critical to the success of the forum case retrieval methods discussed in the previous chapter. Conventional medical information extraction tasks generally aim at extracting finer granularity semantic information units such as entities and relations. However in previous chapter we observed that extraction of such information tends to be difficult in case of noisy and non-technical language found in forum texts and is not very helpful in retrieval applications built on top of them.

In this work, we relax this conventional goal of fine grained extraction and study an easier extraction task where we aim at extracting sentences that belong to a set of predefined semantic categories. That is, we take a sentence as a unit for extraction. Specifically, we study this problem in the context of extracting medical case description from medical forums.

A variety of medical health forums exist online. People use them to post their problems, get advices from experienced patients, get second opinions from other doctors, or merely to vent out their frustration.

Compared with well-structured sources such as Wikipedia, forums are more valuable in the sense that they contain first hand patient experiences with richer information in terms of what treatments are better than others and why. Besides this, on forums, patients explain their symptoms much more freely than those mentioned on relatively formal sources like Wikipedia. And hence, forums are much more easier to understand for a naïve user.

However, even on targeted forums (which focus on a single disease), data is quite unstructured. There is therefore a need to structure out this information and present it in a form that can directly

be used for a variety of other information extraction applications like the collecting of medical case studies pertaining to a particular disease, mining frequently discussed symptoms, identifying correlation between symptoms and treatments, etc.

A typical medical case description tends to consist of two aspects:

- **Physical Examination/Symptoms (PE)**: This covers current conditions and includes any condition that is the focus of current discussion. Note that if a drug causes an allergy, then we consider it as a PE and not a medication. Any condition that is the focus of conversation, i.e. around which treatments are being proposed or questions are being asked is considered PE even if the user is recounting their past experience.

- **Medications (MED)**: Includes medications the person is currently taking, or is intending to take, or any medication on which the question is targeted. Medications do not necessarily mean drugs. Any measures (including avoiding of substances) taken to treat or avoid the symptoms are considered as medication. Sometimes, users also mention other things like constituents of the drug, how much of the drug to consume at a time, how to get access to a medication, how much it costs, side effects of medications, other qualities of medications etc.

Figure 5.1 shows an example of PE and MED labelings.

<MED>i was told hot peppers ie in salsa,
mexican,spicy,szechuan/polynesian type foods are great treatments.</MED>
<PE>They help against nasal/sinusitis/rhinitis conditions.</PE>
<PE>ie allergies/colds</PE>
<MED>also,i believe zyrtec and antihistimines can be and should be
taken before bedtime to eliminate daytime drowsiness.</MED>
<MED>Try vitamin c drops (also aids throat dryness) as a supplement.
Vitamin C can also be found in red peppers.
Peppers can clear passageways i heard in an article recently.</MED>

Figure 5.1: Example of PE and MED labelings

We thus frame the problem of extracting medical case descriptions as extracting sentences that describe any of these two aspects. Specifically, the task is to identify sentences in each of the

two related categories (i.e., PE and MED) from forum posts. As an extraction task, this task is "shallower" than conventional information extraction tasks such as entity extraction in the sense that we extract a sentence as a unit, which makes the extraction task more tractable. Indeed, the task is more similar to sentence categorization. However, it also differs from a regular sentence categorization task (e.g., sentiment analysis) in that the multiple categories are usually closely related and categorization of multiple sentences may be dependent in the sense that knowing the category of one sentence may influence our decision about the category of another sentence nearby. For example, knowing that a sentence is in the category PE should increase our belief that the next sentence is of category of PE or MED.

We solve the problem using two popular machine learning methods, Support Vector Machines (SVM) and Conditional Random Fields (CRF). We define and study a large set of features, including two kinds of novel features: (1) novel features based on semantic generalization of terms, and (2) novel features specific to forums.

Since this is a novel task, there is no existing data set that we can use for evaluation. We thus create a new data set for evaluation. Experiment results show that both groups of novel features are effective and can improve extraction accuracy. With the best configurations, we can obtain an accuracy of up to 75%, demonstrating feasibility of automatic extraction of medical case descriptions from forums.

5.2 Problem formulation

Let $P = (s_1, ...s_n)$ be a sequence of sentences in a forum post. Given a set of interesting categories $C = \{c_1, ..., c_k\}$ that describe a medical case, our task is to extract sentences in each category from the post P. That is, we would like to classify each sentence s_i into one of the categories c_i or *Background*, which we treat as a special category meaning that the sentence is irrelevant to our extraction task. Depending on specific applications, a sentence may belong to more than one category.

In this work, we focus on extracting sentences of two related categories describing a medi-

cal case: (1) Physical Examination (PE), which includes sentences describing the condition of a patient (i.e., roughly symptoms) (2) Medications (MED), which includes sentences mentioning medications (i.e., roughly treatment). These sentences provide a basic description of a medical case and can already be very useful if we can extract them.

We chose to analyze at the sentence level because a sentence provides enough context to detect the category accurately. For example, detecting the categories at word level will not help us to mark a sentence like *"I get very uncomfortable after eating cheese"* as PE or mark a sentence like *"It's best to avoid cheese in that case"* as MED. Here the problem is loosely represented by a combination of *"uncomfortable eating cheese"* and the solution is represented loosely by *"avoid cheese"*. Indeed, in preliminary analysis, we found that most of the times, the postings consist of PE and MED type sentences.

5.3 Methods

We use SVMs and CRFs to learn classifiers to solve our problem. SVMs represent approaches that solve the problem as a classification/categorization task while CRFs solve the problem as a sequence labeling task. In this section, we provide the basics of SVMs and CRFs.

5.3.1 Support vector machines

SVM first introduced in [22], are a binary classifier that constructs a hyperplane which separates the training instances belonging to the two classes. SVMs maximize the separation margin between this hyperplane and the nearest training datapoints of any class. The larger the margin, the lower the generalization error of the classifier. SVMs have been used to classify both linearly and non-linearly seperable data, and have been shown to outperform other popular classifiers like decision trees, Naïve Bayes classifiers, k-nearest neighbor classifiers, etc. We use SVMs as a representative classifier that does not consider dependencies between the predictions on multiple sentences.

5.3.2 Conditional random fields

Each of the sentences in the postings can itself contain features which help us to categorize it. Besides this, statistical dependencies exist between sentences. Intuitively, a MED sentence will follow a PE sentence with high probability, but the probability of a PE sentence following an MED sentence would be low. Conditional random fields are graphical models that can capture such dependencies among input sentences. A CRF model defines a conditional distribution $p(y|x)$ where y is the predicted category (label) and x is the set of sentences (observations). CRF is an undirected graphical model in which each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. The observation x can be dependent on the current hidden label y, previous n hidden labels and on any of the other observations in a n order CRF. CRFs have been shown to outperform other probabilistic graphical models like Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MeMMs). Sutton and McCallum sutton06introduction provide an excellent tutorial on CRFs.

5.4 Features

To perform our categorization task, we use the following features.

- **Word based features**: This includes unigrams, bigrams and trigrams in the current sentence. Each of the n-grams is mapped to a separate boolean feature per sentence where value is 1 if it appears in sentence and 0 otherwise.

- **Semantic features**: This includes Unified Medical Language System (UMLS[1]) semantic groups of words in the current sentence. UMLS is a prominent bio-medical domain ontology. It contains approximately a million bio-medical concepts grouped under 135 semantic groups. MMTX[2] is a tool that allows mapping of free text into UMLS concepts and groups. We use these 135 semantic groups as our semantic features. In order to generate these features, we first process this sentence through MMTX API which provides all the semantic

[1] http://www.nlm.nih.gov/research/umls/
[2] http://mmtx.nlm.nih.gov/

groups that were found in the sentence. Each of the semantic groups becomes a boolean feature.

- **Position based features**: We define two types of position based features: position of the current sentence in the post and position of the current post in the thread. These features are specific to the forum data. We include these features based on the observations that first post usually contains condition related sentences while subsequent posts often contain treatment measures for the corresponding condition. Each of the position number of a sentence in a post and a post in a thread is mapped to a boolean feature which gets fired for a sentence at a particular position. E.g. For a sentence at position i in a post, POSITION_IN_POST_i would be set to 1 while other features POSITION_IN_POST_j where $j \neq i$ would be set to 0.

- **User based features**: We include a boolean feature which gets fired when the sentence is a part of a post by the thread creator. This feature is important because most of the posts by a thread creator have a high probability of being a PE.

- **Tag based features(Edge features)**: We define features on tags (PE/MED/Backgnd) of previous two sentences to capture local dependencies between sentences. E.g., a set of medication related tags often follow a description of a condition. We use these features only for CRF based experiments.

- **Morphological features**: These include one boolean feature each for presence of
  - a capitalized word in the sentence
  - an abbreviation in the sentence
  - a number in the sentence
  - a question mark in the sentence
  - an exclamation mark in the sentence

- Length based features: We also consider the number of words in a sentence as a separate type of feature. Feature LENGTH_i becomes true for a sentence containing i words.

Category	Labeler 1	Labeler 2
PE	513	517
MED	286	280
Background	695	697

Table 5.1: Labeling results

5.5 Experiments

5.5.1 Dataset

Evaluation of this new extraction task is challenging as no test set is available. To solve this problem, we opted to created our own test set. HealthBoards[3] is a medical forum web portal that allows patients to discuss their ailments. We scraped 175 posts contained in 50 threads on allergy i.e., an average of 3.5 posts per thread and around 2 posts per user with a maximum of 9 posts by a particular user. Two humans were asked to tag this corpus as conditions (i.e., PE category) or treatments (i.e., MED category) or none on a per sentence basis. The corpus consists of 1494 sentences. Table 5.1 shows the labeling results. The data set is available at (http://timan.cs.uiuc.edu/downloads.html). Also the labeling results match quite well (82.86%) with a Kappa statistic value of 0.73. Occasionally (around 3%) PE and MED both occur in the same sentence and the labelers chose to mark such sentences as PE. In the case when the two labelers disagree, we manually analyzed the results and further chose one of them for our experiments.

5.5.2 Evaluation methodology

For evaluation, we use 5-fold cross validation. For CRFs, we used the Mallet[4] toolkit and for SVM, we used SVM-Light[5]. We experimented by varying the size of the training set, with different feature sets, using two machine learning models: SVMs and CRFs. Our aim is to accurately classify any sentence in a post as PE or MED or background. First we explore and identify the feature sets that help us in attaining higher accuracy. Next, we identify the setting (sequence labeling by CRFs or independent classification by SVMs) that works better to model our problem.

[3] http://www.healthboards.com
[4] http://mallet.cs.umass.edu/
[5] http://svmlight.joachims.org/

We present most of our results using four metrics: precision, recall, F1 measure and average accuracy which is the ratio of correctly labeled sentences to the total sentences.

We considered the following features: all the 2647 words in the vocabulary (no stop-word removal or any other type of selection), 10858 bigrams, 135 semantic groups from UMLS, two position based features, one user based feature, two tag based features, four morphological features and one length based feature as described in the previous section. Thus our feature set is quite rich. Note that other than the usual features, semantic, position-based and user-based features are specific to the medical domain or to forum data.

5.5.3 Basic results

First we considered word features, and learned a linear chain CRF model. We added other sets of features one by one, and observed variations in accuracy. Table 5.2 shows the accuracy in terms of precision, recall and F1. Note that these results are for an Order 1 linear-chain CRF. Accuracy is measured as ratio of the number of correct labelings of PE, MED and background to the total number of sentences in our dataset. Notice that the MED accuracy values are in general quite low compared to those of PE. As we will discuss later, accuracy is low for MED because our word-based features are not discriminative enough for the MED category.

From Table 5.2, we see that the accuracy keeps increasing as we add semantic UMLS based features, position based features and morphological features. However, length based features (word count), user-based faetures, and bigrams do not result in any improvements. We also tried trigrams, but did not observe any accuracy gains. Thus we find that semantic features and position-based features which are specific to the medical domain and the forum data respectively are helpful when added on top of word features, while generic features such as length-based features tend to not add value.

We also trained an order 2 CRF using the same set of features. Results obtained were similar to order 1 CRFs and so we do not report them here. This shows that local dependencies are more important in medical forum data and global dependencies do not add further signal.

Further, we perform experiments using SVMs using the same set of features. Table 5.3 shows

Feature set	PE Prec	MED Prec	PE Recall	MED Recall	PE F1	MED F1	Accuracy %
Word	0.60	0.49	0.65	0.36	0.62	0.42	63.43
+Semantic	0.61	0.52	0.68	0.37	0.64	0.43	65.05†
+Position	0.63	0.54	0.7	0.34	0.66	0.42	65.45
+Morphological	0.64	0.52	0.69	0.36	0.66	0.42	65.70
+WordCount	0.62	0.51	0.70	0.33	0.66	0.40	65.23
+Thread Creator	0.62	0.51	0.71	0.34	0.66	0.41	65.49
+Bigrams	0.62	0.51	0.69	0.34	0.66	0.41	64.82

Table 5.2: Order 1 Linear Chain CRF. †Improvement over only word features significant at 0.05-level, using Wilcoxon's signed-rank test

accuracy results on SVM. Again PE is detected with higher accuracy compared to MED. Unlike CRFs, SVMs do not incorporate the notion of local dependencies between sentences. However, we observe that SVMs outperform CRFs, as is evident from the results in Table 5.3. This is interesting, since it suggests that the SVM accuracy can potentially be further enhanced by incorporating such dependency information (e.g. in the form of new features). We leave this as part of future work.

Feature set	PE Prec	MED Prec	PE Recall	MED Recall	PE F1	MED F1	Accuracy %
Word	0.65	0.52	0.71	0.28	0.68	0.36	66.13
+Semantic	0.73	0.54	0.73	0.38	0.73	0.45	71.02†
+Position	0.71	0.52	0.71	0.35	0.71	0.42	69.61
+Morphological	0.72	0.53	0.72	0.38	0.72	0.44	70.28
+WordCount	0.74	0.54	0.72	0.37	0.73	0.44	71.55
+Thread Creator	0.74	0.56	0.72	0.39	0.73	0.46	72.02
+Bigrams	0.75	0.54	0.72	0.40	0.74	0.46	71.69

Table 5.3: SVM results. †Improvement over only word features significant at 0.05-level, using Wilcoxon's signed-rank test

Figure 5.2 shows an example of a forum post (which talks about allergy to dogs) being tagged using our CRF model.

<BKG>lori-lynn , </BKG>
<PE>you said he does well with the poms , but you also said he takes shots, so i wondered if the shots were for dog allergies</PE>
<PE>a lot of his friends have dogs , though , and he ' s so very allergic that he has trouble at their homes .</PE>
<MED>we opted not to go with the shots . </MED>
<BKG>i ' m still a little leary about adopting a dog . </BKG>
<BKG>i would just hate it if we did have reactions , because i know we ' d bond with the dog very quickly . </BKG>

Figure 5.2: Tagging example of a forum post

Classifier	PE Prec	PE Recall	PE F1	MED Prec	MED Recall	MED F1	Accuracy %
SVM (all* features)	0.72	0.53	0.72	0.38	0.72	0.44	70.28
SVM (selected features)	0.75	0.75	0.75	0.61	0.33	0.44	75.08†
CRF (all* features)	0.64	0.52	0.69	0.36	0.66	0.42	65.70
CRF (selected features)	0.60	0.77	0.67	0.58	0.37	0.45	65.93†

Table 5.4: Accuracy using the best feature set. (*Word +Semantic +Position +Morphological features). †Improvement over all* features significant at 0.05-level, using Wilcoxon's signed-rank test

5.5.4 Feature selection

Incremental addition of different feature types did not lead to substantial improvement in performance. This suggests that none of the feature classes contains all "good" features. We therefore perform feature selection based on information gain and choose the top 4253 features from among all the features discussed earlier, based on a threshold for the gain. This results in improvement in the accuracy values over the previous best results (Table 5.4).

Among the word feature set, we found that important features were *allergy, alergies, food, hives, allergic, sinus, bread*. Among bigrams, *allergic_to, ear_infections, my_throat, are_allergic, to_gluten, food_allergies* have high information gain values. Among the UMLS based semantic groups, we found that *patf (Pathologic Function), dsyn (Disease or Syndrome), orch (Organic Chemical), phsu (Pharmacologic Substance), sosy (Sign or Symptom)* have high information gain values. Also looking at the word count feature, we notice that background sentences are generally short sentences. All these features are clearly highly discriminative.

5.5.5 Variation in training data size

We varied the amount of training data used for learning the models to observe the variation in performance with size of training data. Table 5.5 shows the variation in accuracy (PE F1, MED F1 and average accuracy) for different sizes of training data using CRFs. In general, we observe that accuracy improves as we increase the training data, but the degree varies with the feature sets used. We see similar trends in SVM also. These results show that it is possible to further improve prediction accuracy by obtaining additional training data.

Feature set	25%	50%	75%	100%
Word	0.59/0.21/0.57	0.6/0.36/0.60	0.61/0.39/0.62	0.62/0.42/0.63
+Semantic	0.61/0.17/0.59	0.63/0.32/0.61	0.64/0.38/0.63	0.64/0.43/0.65
+Position	0.59/0.18/0.56	0.64/0.29/0.60	0.65/0.33/0.62	0.66/0.42/0.65
+Morphological	0.6/0.19/0.57	0.64/0.32/0.61	0.65/0.37/0.63	0.66/0.42/0.65
Best	0.61/0.18/0.65	0.66/0.28/0.64	0.66/0.38/0.66	0.69/0.43/0.68

Table 5.5: PE F1, MED F1 and Average Accuracy for various sizes of training data set.

5.5.6 Probing into the low MED accuracy

As observed in Tables 5.2 and 5.3, MED accuracy is quite low compared to PE accuracy. We wish to gain a deeper insight into why the MED accuracy suffers. Therefore, we plot the frequency of words in sentences marked as PE or MED versus the rank of the word as shown in the figure 5.3. We removed the stop words. Observe that for PE the curve is quite steep. This indicates that there are some discriminative words which have very high frequency and so the word features observed in the training set also get fired for sentences in the test set with high probability. While for MED, we observe that most of the words have very low frequencies. This basically means that discriminative words for MED may not occur with good enough frequency. So, many of the word features that show up in the training set may not appear in the test data. Hence, MED accuracy suffers.

5.5.7 Multi-class vs single class categorization

Classifier Type	PE Prec	PE Recall	PE F1	MED Prec	MED Recall	MED F1
SVM PE vs BKG	0.79	0.64	0.71	-	-	-
SVM MED vs BKG	-	-	-	0.6	0.28	0.39
SVM Multi-class	0.73	0.73	0.73	0.54	0.38	0.45
CRF PE vs BKG	0.68	0.64	0.66	-	-	-
CRF MED vs BKG	-	-	-	0.53	0.3	0.39
CRF Multi-class	0.61	0.68	0.64	0.52	0.37	0.43

Table 5.6: Multi-class vs Single-class categorization with word+semantic features

Note that our task is quite different from plain sentence categorization task. We observe that there is a dependence between the categories (PE/MED) that we are trying to predict per sentence. For example, considering 100% training data, Table 5.6 compares the precision, recall and F1 values when SVM and CRF are trained as single class classifiers using word+semantic features with

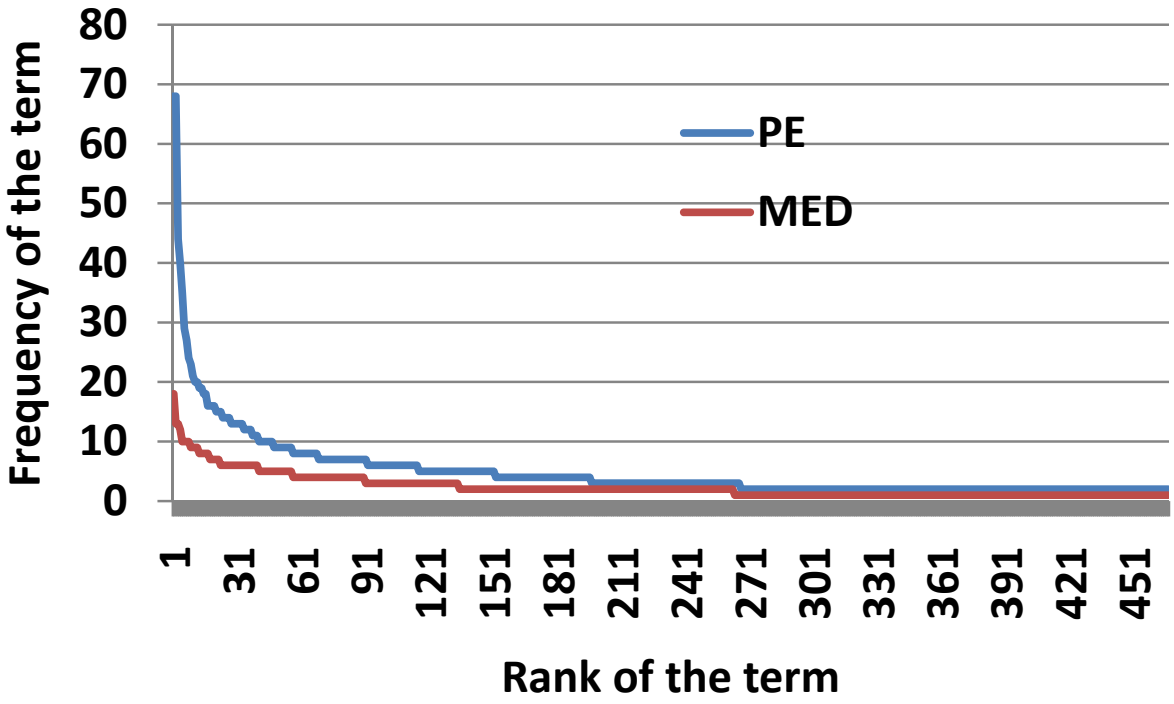

Figure 5.3: Freq of words vs rank for PE and MED

the multi-class results obtained previously. Results are generally better when we do multi-class categorization versus single-class categorization. This trend was reflected for other featuresets also.

5.5.8 Analysis of transition probabilities

Table 5.7 shows the transition probabilities from one category to another as calculated based on our labelled dataset. BOP is beginning of posting and EOP is end of posting. Note that posts often start with a PE or a background sentence and often end with a background sentence. Also, consecutive sentences within a posting tend to belong to the same category.

5.5.9 Error analysis

We also perform some error analysis on results using the best feature set. Table 5.8 shows the confusion matrix for CRF/SVM. We observe many of the MED errors are because an MED sen-

	PE	MED	Backgnd	EOP
PE	0.54	0.13	0.28	0.05
MED	0.15	0.51	0.30	0.04
Backgnd	0.18	0.08	0.54	0.20
BOP	0.40	0.07	0.53	0.0

Table 5.7: Transition probability values

	PE	MED	Backgnd
PE	424/404	37/37	81/101
MED	102/70	107/95	81/125
Backgnd	164/62	55/21	618/754

Table 5.8: Confusion matrix showing counts of actual vs predicted labels for (Best CRF Classifier/Best SVM Classifier)

tence often gets marked as PE. This basically happens because some sentences contain both PE and MED. Other than that some of the PE keywords are also present in MED sentences, and since the few discriminative MED keywords are quite low in frequency, MED accuracy suffers. E.g. The sentence *"i'm still on antibiotics for the infection but they don't seem to be doing any good anymore."* was labeled as MED but marked as PE by the CRF. The sentence clearly talks about a medication. However, the keyword *"infection"* is often observed in PE sentences and so the CRF marks the sentence as PE.

5.6 Discussion

In this chapter, we studied a novel shallow semantic information extraction task where the goal was to extract sentences relevant to a predefined set of categories that describe a medical case. By relaxing the constraint of finding precise entities at word level, and restricting ourselves to only three semantic classes, our novel problem has become more tractable. As a result using our proposed supervised learning approaches based on CRF and SVM, we were able to train fairly accurate models with relatively small training data. Our results show that (1) the proposed new features based on generalized terms and forum structure are effective in improving the extraction accuracy, and (2) it is feasible to automatically extract medical cases in this way, with the best prediction accuracy above 75%.

The task was crucial to the development of forum based autonomous agents as it allowed us

to separate case related sentences in the query from the background. Sentence level extraction also allowed us to identify keywords that are not standard medical entities, but are nevertheless important to the case query. The benefits of using such a method for similar case-based retrieval on forum data have already been shown in the previous chapter.

Our work can be further extended in several ways. First, since constructing a test set is labor-intensive, we could only afford experimenting with a relatively small data set. It would be interesting to further test the proposed features on larger data set. Second, while in CRF, we have shown adding dependency features improves performance, it is unclear how to evaluate this potential benefit with SVM. Since SVM generally outperforms CRF for this task, it would be very interesting to further explore how we can extend SVM to incorporate dependency.

Chapter 6

Automated Resolution of Healthcare Community Questions

6.1 Introduction

In this chapter, we study the third application scenario where an agent proactively helps resolve generic healthcare queries found on community question answering (cQA) websites by extracting text snippets from health information webpages. Community QA (cQA) websites such as Yahoo! Answers, are highly popular. However, many questions are either left unanswered, partially answered or are answered with a significant delay. Past research has tackled this problem by finding questions from the archive, similar to the one posed by the user [42]. In many cases however, similar questions may not already exist or may be hard to find.

In this work we propose a novel and complementary solution. We believe that many of the questions may be resolved via online knowledge-base websites such as wikipedia[1] or eMedicinehealth[2]. The following example shows that a question about lowering blood pressure can be potentially automatically resolved by generating a response using a paragraph from eMedicineHealth:

Question: *What is a good way to lower blood pressure? I would like to go off of my meds if possible. I would like to know the right types of foods and drinks and self care i can do. Thanks in advance.*

Sample Response from eMedicineHealth: *Lifestyle changes are important to help control high blood pressure. Even if your doctor has prescribed medicine for you, you can still take many steps at home to lower your blood pressure and reduce your risk. Some people can even take less*

[1] http://www.wikipedia.org
[2] http://www.emedicinehealth.com

medicine after making these changes. Make these lifestyle changes to help lower your blood pressure:Lose extra weight,Eat healthy foods....

Such knowledge-base websites tend to follow a standard structure. Each webpage discusses different aspects of a particular entity. For example an article on Bronchitis[3] on emedicinehealth, discusses the various aspects of the disease entity "Bronchitis", like "causes", "symptoms", "treatment", "prognosis" etc. Moreover, each aspect generally appears as a section heading (eg. "Bronchitis Treatment") followed by its text description.

One can view this data as being organized in a relational database with schema represented by the section headings. For example a relation with attributes "(Disease,Symptoms)" will store tuples of the form "(Bronchitis,Text describing symptoms of Bronchitis)". Note that while the values stored in the database are predominantly verbose text descriptions, there are no restrictions on the kind of information that can be stored. For example one could define more complex relations that store numeric values eg. "(Disease,Drug,Dosage)", where the attributes "Disease" and "Drug" store text values while "Dosage" is numeric.

Our goal in this work is to respond to an unresolved question by mining the most suitable text value from the database. We will subsequently refer to such a domain specific relational database as a knowledge-base to distinguish it from a regular database. We will also refer to this novel problem of resolving questions by mining text values from it, as knowledge-based Question Resolution (kbQR). We will also frequently use the terms "document", "value" and "database value" interchangeably to refer to a specific atomic value in the knowledge-base.

The kbQR is a novel text mining problem which has clear difference from the existing information retrieval and question answering tasks. First, it differs from regular text retrieval in that rather than retrieving documents based only on keyword/semantic similarity, the presence of relations between text values also offers us the opportunity to perform limited "reasoning" via sql queries. Thus the challenge in kbQR is to identify relevant sql queries that can help generate a response to the question. In this regard our problem is closer to research in Natural Language Interfaces (NLI),

[3] http://www.emedicinehealth.com/bronchitis/article_em.htm

since both attempt to return values from a database [65, 60, 83]. However the nature of questions posed, databases used and the responses expected are all quite different from those in our case. In NLI applications, the natural language input from a user are often short and precise commands or requests making them relatively easy to parse. The databases also generally contain precise values which are easy to match with the keywords in the input command. Consequently, the methods used in NLI involve parsing the question text and directly trying to infer an underlying well-formed relational database query. In contrast, questions posted on community QA websites are uniformly real questions, and tend to be long, noisy and verbose, with a lot of background information, which makes NLI parsing methods unsuitable. Moreover our goal is not to find a unique sql query. Infact since the knowledge-base values are also verbose, the question will partially match many different values and there will likely be multiple sql queries capable of generating a response.

To solve this novel text mining problem, we propose a general probabilistic framework that would generate the response in two steps. First we generate candidate sql queries along with a confidence score on their likelihood of generating the response. We then execute the high scoring queries to generate candidate response values. The final score of each candidate value is based on the confidence scores of all queries that generated it.

We further use the framework to derive a specialized model for responding to questions on cQA websites, that takes into account the noisy descriptive questions and verbose values. In particular, we show how the set of relevant sql queries may be mined, and the parameters of the model estimated, by automatically generating a training set from questions already available on cQA websites.

We evaluate our model by building a kbQR system for responding to healthcare questions on Yahoo! answers, using wikipedia as the knowledge-base. Results show that it is indeed feasible to resolve cQA questions using our method.

To summarize, this work makes the following major contributions:

1. We introduce and study the novel knowledge-based Question Resolution (kbQR) problem, where online knowledge databases are leveraged to automatically resolve or facilitate resolu-

tion of unresolved questions posted on community QA websites.

2. We propose a general probabilistic framework for solving the kbQR problem and show how it may be instantiated to build a QA system for web communities.

3. We show how relevant queries may be mined and model parameters estimated using an automatically generated training set.

4. We evaluate the proposed framework in healthcare domain and demonstrate the proposed new strategy of kbQR and effectiveness of the proposed QA algorithms.

6.2 Problem definition

In this section we provide a formal definition of the kbQR task and the notations that will be used in subsequent sections.

Problem: Given a knowledge database D and a question q, our goal is to return a database value v_a as the response.

Input: There are two input variables q and D.

q may be (semi-)structured or unstructured. While our proposed framework does not restrict the nature of q, questions encountered in community QA websites tend to be predominantly unstructured.

The database D comprises a set of relations $R = \{r_1, r_2, ..., r_n\}$. The schema of each r_i is represented via the set of attributes $A_i = \{a_{i1}, a_{i2}, ..., a_{iK_i}\}$. Each attribute a_{ij} represents the jth attribute in the ith relation and is considered unique. Finally we define $A_D = \cup_{i=1}^{n} A_i$ as the set of all database attributes and use $a \in A_D$ to represent some attribute in D. For example in a knowledgebase with two relations $R = \{Disease\_Treatment, Disease\_Symptoms\}$ with attributes $(Disease, Treatment)$ and $(Disease, Symptoms)$ we get

$$
\begin{aligned}
r1 &= Disease\_Treatment \\
A_1 &= \{Disease\_Treatment.Disease, Disease\_Treatment.Treatment\} \\
a_{11} &= Disease\_Treatment.Disease \\
a_{12} &= Disease\_Treatment.Treatment \\
r2 &= Disease_Symptoms \\
A_2 &= \{Disease\_Symptoms.Disease, Disease\_Symptoms.Symptoms\} \\
a_{21} &= Disease\_Symptoms.Disease \\
a_{22} &= Disease\_Symptoms.Symptoms \\
A_D &= A_1 \cup A_2 \\
&= \{Disease\_Treatment.Disease, Disease\_Treatment.Treatment, \\
&\quad Disease\_Symptoms.Disease, Disease\_Symptoms.Symptoms\}
\end{aligned}
$$

Note that even though the $Disease\_Treatment.Disease$ and $Disease\_Symptoms.Disease$ have the same semantic interpretation, we treat them as two different and unique attributes. Consequently the total attributes in a database is just the sum of number of attributes in individual relations i.e. $\sum_{i=1}^{n} |A_i| = |A_D|$.

A database value v_{ijk} represents the atomic value appearing in the kth tuple of the ith relation under the j attribute. We define $Attr(v) \in A_D$ to represent the attribute of the value v i.e. for $v = v_{ijk}$, $Attr(v_{ijk}) = a_{ij}$. Finally we use V_D to represent the set of all knowledge-base values.

Output: The expected output is a value $v_a \in V_D$ such that v_a forms a plausible response to q. We assume the question is resolvable through a single value in the database.

We view the core computation problem as that of ranking knowledge-base values based on their probability of being the response and return the one that scores the highest. Alternatively depending upon the application needs and probability scores, we can also return more than one or no values as well.

6.3 A general probabilistic framework for kbQR

Intuitively, our approach involves the following steps:

1. First we identify values in the knowledge-base that are similar to the question. These values can be considered as information the user has already provided.
2. Next we "reason" with these values by incorporating them as constraints in sql queries. The knowledge-base values returned by executing these queries become candidate responses.
3. Finally we rank the values returned by the sql queries based on their probability of being the response.

Consider an example in the healthcare domain. For simplicity we assume, our knowledge-base only contains a single relation Rel with three attributes (Disease, Symptoms, Treatment), and stores values for only two diseases i.e. two tuples $\{(dis1, symp1, treat1), (dis2, symp2, treat2)\}$. Now let a user asks a question describing a set of symptoms and expects a treatment description in response. In the first step we match the question to all values stored in the knowledge-base. Each of the 6 values in the knowledge-base will have some similarity score, with the symptom descriptions $symp1$ and $symp2$ likely having the highest. Consequently, a subset of queries we can execute will be (here we refer to $symp1$ and $symp2$ as constraints)

`select Symptoms from` Rel `where Symptoms =` $symp1$
`select Symptoms from` Rel `where Symptoms =` $symp2$
`select Treatment from` Rel `where Symptoms =` $symp1$
`select Treatment from` Rel `where Symptoms =` $symp2$

The first two queries return the matched symptom value itself. This is equivalent to the behavior of a traditional retrieval method. The next two queries will return the treatment text value corresponding to the matched symptom. These queries are more likely to retrieve the response the user expected. Our confidence in the query to be executed will depend on a) how relevant the

query is to the question and b) how well its constraint matches the question.

In a more general case, we may have to consider 100s of matched knowledge-base values as constraints and potentially 1000s of candidate queries of arbitrary complexity. This significantly complicates the problem of finding the most relevant response value. In subsequent discussion we present a principled way of approaching the problem through a general probabilistic framework for modeling the uncertainties.

We begin with the conditional probability $P(V = v|Q = q)$, which is the probability that a value v in the knowledge-base is the response to the question q. The best response v_a would then be given by maximizing over the set of all possible values in the database V_D:

$$v_a = \arg max_{v \in V_D} P(V = v|Q = q)$$

We now decompose this expression into two conditional probabilities "bridged" by a sql query that can potentially capture the semantics of the user's question q. Formally, let S_D be the set of legitimate queries that one is allowed to execute on the database D, then the probability of a value v may be obtained by marginalizing over it:

$$P(V = v|Q = q) = \sum_{s \in S_D} P(V = v|S = s, Q = q)P(S = s|Q = q)$$

This decomposition has a quite meaningful interpretation: $P(S = s|Q = q)$ captures the uncertainty in inferring what query to execute for question q. Since a query could potentially return multiple knowledge-base values, $P(V = v|S = s, Q = q)$ captures the uncertainty in deciding the response among the returned values. In cases where a query only returns a single value $P(V = v|S = s, Q = q)$ trivially becomes 1. On the other hand if v is not among the values generated by s, it is 0. To keep the notation simple, in subsequent text, unless otherwise necessary we will drop the random variables and simply write for example $P(S = s|Q = q)$ as $P(s/q)$.

In theory the set S_D could encompass all possible queries executable on the knowledge-base. However we want to restrict it to only those queries which we feel are relevant in resolving questions. To this end, we can make additional simplifying assumptions. We will restrict S_D to only

queries that have a single target attribute and use a single value as constraint. The first assumption is natural since we are trying to retrieve a single value. As we will see later, the second assumption is also not particularly restrictive.

A sql query is a complex object which can be encoded using many different features such as "constraint used","target attribute", "number of relations touched" etc. We will encode a query using two features - its constraint and the target attribute i.e.

$$P(v|q) = \sum_{s \in S_v \subset S_D} P(v|s,q)P(Cons(s), Att(s)|q)$$

Where $Cons(s) \in V_D$ is the constraint used in query s, $Att(s) \in A_D$ is its target attribute and $S_v \subset S_D$ is the set of queries that generate the value v. Assuming that the inference of target attribute and the constraint given a question are independent, we can further simplify the equation as:

$$P(v|q) = \sum_{s \in S_v \subset S_D} P(v|s,q)P(Cons(s)|q)P(Att(s)|q)$$

Finally since any candidate response value appears only under a single attribute in the knowledge-base, assuming $Att(v)$ to be the attribute of v, we can write the final ranking function as

$$\begin{aligned} P(v|q) &= P(Att(v)|q) \sum_{s \in S_v} P(v|s,q)P(Cons(s)|q) \\ logP(v|q) &= log(P(Att(v)|q)) + log(\sum_{s \in S_v} P(v|s,q)P(Cons(s)|q)) \end{aligned}$$

Note from the equation that the distribution over the target attribute and the constraint appear as separate components in the log. This is a major benefit of the model. It means that when maximizing the log likelihood over some training data, we can estimate the parameters of these two distributions independently.

Also note that the distribution over constraints, appears in a summation. This means that while calculating $P(v|q)$ we sum over all constraints that can be used in queries to generate v. Thus restricting S_D to only single constraint queries, still allows v to receive contributions from all relevant constraints.

Finally the equation suggests that in order to build a kbQR system, one needs to instantiate the following components

Legitimate Query Set: S_D defines the set of legal queries which the system is allowed to execute to retrieve a response. For most large knowledge-bases, this will have to be automatically inferred by mining queries from some collection of known QA pairs.

Constraint Prediction Model: $P(Cons(s)|q)$ captures our uncertainty on whether some value in the knowledge-base can be used as a constraint for the question. Note that $Cons(s) \in V_D$. Hence it is a distribution over all knowledge-base values, conditioned on the question.

Attribute Prediction Model: $P(Att(v)|q)$ captures our uncertainty on the attribute of the value expected in the response. It is a distribution over all possible attributes $Att(s) \in A_D$, given a question. Its effect is similar in spirit to the question type detection that generally forms an initial step in most QA systems [41].

Value Prediction Model: $P(v|s,q)$ allows us to favor some values over others based on question features, among all values generated by query s. For example, we can use it to ensure some degree of textual similarity between the response and the question. When no such preferences exist, the distribution can simply be made uniform. For this work we will assume the distribution is uniform i.e.

$$P(v|s,q) = \frac{1}{|Val(s)|}$$

where $Val(s)$ is the set of all values generated on executing the query s.

In the next section we describe how the different components may be instantiated to build a QA system for community QA websites.

6.4 kbQR for web communities

In this section we discuss a possible instantiation of general kbQR suitable for web communities and show how its parameters may be estimated. We discuss all components except query mining which is discussed later in section 6.5.2 after describing the knowledgebase.

6.4.1 Constraint distribution ($P(Cons(s)|q)$)

Since both questions and the database values tend to be verbose, one cannot ascertain the existence of a database value merely by looking at whether a question contains it. The constraint model instead needs to incorporate an intelligent similarity function between the question and a database value. In addition, questions tend to contain a number of discourse related background keywords such as "thanks in advance" etc., which we need to filter out. We define the probability of a knowledge-base value $v \in V_D$ of being a constraint for a question q to be

$$P(Cons(s)|q) = \frac{e^{\alpha Sim(Cons(s),q)}}{\sum_{v' \in V_D} e^{\alpha Sim(v',q)}}$$

Where for some v, $Sim(v, q)$ denotes textual similarity between v and q calculated by an intelligent similarity function and α is a parameter. $Sim(v, q)$ is defined by treating the task of finding a matching database value as a retrieval problem, with the question q as a query and a value v as the document. We use the KL-Divergence retrieval model [96] for scoring relevance of values.

$$Sim(v, q) = \sum_{w \in Vocab} P(w|\theta_q) log P(w|\theta_v)$$

where θ_q and θ_v are multinomial word distributions for question and value respectively.

The value model θ_v characterized as the word distribution $P(w|\theta_v)$ is estimated using Dirichlet

prior smoothing over the collection [96]:

$$P(w|theta_v) = \frac{c(w,V) + \mu P(w|V_D)}{|D| + \mu}$$

where $c(w, D)$ is the count of word w in the document D, $P(w|V_D)$ is the probability of w in the entire collection. Optimal value of μ is generally set at 4800.

To remove the background keywords while estimating the question model θ_q, we treat the question as being generated from a two component mixture model, with a fixed background model θ_{C_Q} which represents the probability of a word w in a large collection of questions C_Q. Probability of a word in a question is the given by

$$P(w) = (1 - \lambda)P(w|\theta_q) + \lambda P(w|\theta_{C_Q})$$

The model θ_q can then be estimated using Expecation Maximization as in [97]. In effect, the estimated $P(w|\theta_q)$ would favor discriminative words and give them high probabilities.

6.4.2 Attribute distribution ($P(Att(v)|q)$)

Attribute prediction can be viewed as a multi-class classification task over question features. We therefore model the distribution $P(Att(v)|q)$ with $Att(v) \in A_D$ being the target attribute, using a maximum entropy model.

$$P(Att(v)|q) = \frac{e^{w^T_{Att(v)} \cdot q_F}}{\sum_{a \in A_D} e^{w^T_a \cdot q_F}}$$

where w_a is the weight vector for attribute a and q_F is the vector of question features.

Question features q_F are defined over n-grams (for $n = 1$ to 5) and information gain is used to select top $25K$ most informative features. Parameters for the attribute prediction model are learnt using the L-BFGS [25] method using the Mallet [4] toolkit. A default gaussian prior with

[4] http://mallet.cs.umass.edu/

interpolation parameter set to 1 is used for regularization.

Based on our definition of unique attributes in section 6.2, we treat any two attributes appearing in different relations as different classes for the attribute distribution. For some schema, it may be the case that attributes across two relations are semantically the same. While we do not encounter this case in our knowledge base, in such scenarios one can group together all such attributes into a single attribute class. Alternatively if we know for sure that certain attributes are unlikely to contain a response value, we can simply ignore them.

6.4.3 Response ranking

Once the set of queries S_D and all component models are estimated, the final ranking algorithm for a new question q works as follows:

1. Constraint Selection: Rank all knowledge-base values based on $sim(v, q)$. Select the top-N values to be used as constraints. We set $N = 10$ for all our experiments.

2. Attribute Selection: Generate features for q and use the attribute prediction model to assign scores to every target attribute.

3. Query Selection: Find all queries in S_D containing one of the selected constraints and a target attribute with a non-zero probability and execute them.

4. Response Selection: Score each candidate value v by summing over S_v^e the set of all queries that were executed and generated v

$$score \quad = \quad w_{Att(v)}^T . q_F + log(\sum_{s \in S_v^e} \frac{e^{\alpha Sim(Cons(s), q)}}{|Val(s)|})$$

5. Return the highest scoring value as the response.

6.4.4 Training set generation

In order to mine the queries and estimate the parameters of the attribute and constraint distributions, we need to construct a training set containing questions and their suitable responses from the knowledge-base. This is achieved by leveraging a collection of $80K$ existing healthcare questions threads from Yahoo! Answers website. For any question q, let $\{a_1, a_2...a_n\}$ be the answers provided for it by the users. We want to find a knowledge-base value $v_a \in V_D$ which is most similar to the user provided answers and treat that as the response.

More specifically, we use the KL-divergence retrieval model, the same function used for constraint selection, to generate a ranking of values for each answer a_i. Just like the questions, answers also tend to be noisy and may at times be completely irrelevant. Hence we learn the answer model θ_{a_i} in the same manner as we learnt θ_q in constraint prediction, using a background answer model θ_{C_A} over a large collection of answers. We ignore all answers with less than 30 words to ensure quality.

Once the rankings for each answer are generated, we now need to aggregate them. For each retrieved value, we pick the the K best ranks assigned to it by the answers and average them to generate the final score. The value with the lowest score is labeled as the response. In order to ensure that we only generate training examples for which we are confident, we also reject an instance if:

1. A question has less than K answers available.
2. The lowest scoring value is not ranked at the top by at least two answers.
3. Either two or more values end up with the same lowest score

The parameter K was tuned using a small fully judged validation set discussed in section 6.7.2.

6.5 Knowledgebase construction and query mining

In this section we discuss how we used wikipedia to construct our healthcare knowledgebase and subsequently mined queries to define our legitimate query set S_D.

6.5.1 Wikipedia knowledgebase

We used wikipedia to build our knowledge base. We chose wikipedia because it covers a wide range of healthcare related information, in a language that tends to be at the same level of sophistication as a naive user asking questions on a cQA website. In addition wikipedia is comprehensive enough to be able to resolve many questions raised by users. Finally the strategy used for building the healthcare knowledge base from wikipedia could be used to also build knowledge bases for other domains.

However since there is no fixed set of well defined attributes known beforehand. As a result some effort needs to be put in, a) Identifying relevant domain specific wikipedia pages and b) Converting the section headings into attributes. In this section we detail the process we followed in converting the raw wikipedia pages into a semi-structured knowledge base.

Identifying the domain related wiki pages:

Most wikipedia articles are assigned one or more high level category tags which loosely identify the high level topic of the page. The category keywords are further organized into a hierarchy based on the generality and "is-a" relationships. For example the category "Health" covers 33 subcategories such as "Diseases and Disorders", "health Law" etc. each of which themselves contain multiple sub-categories. More details regarding wikipedia categories are available here[5]. CatScan[6] is a tool that allows users to browse through the wikipedia category trees.

In all we identified categories covering a total of $29K$ wikipedia pages on healthcare related entities. These spanned pages related to Diseases, treatments, medical techniques, drugs etc.

In order to extract these pages we downloaded the entire english wikipedia dump[7] and processed it using the WikipediaExtractor python script available at[8]. The script allowed us to parse the wiki notation and ultimately obtain plain text with some xml markup identifying section headings.

[5] http://en.wikipedia.org/wiki/Wikipedia:Categorization
[6] http://meta.wikimedia.org/wiki/CatScan
[7] http://en.wikipedia.org/wiki/Wikipedia:Database_download
[8] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

Identifying the attributes:

Once the domain relevant pages were identified, our next goal was to identify relations and attributes from the semi-structured wikipages. The section headings in wiki pages are good for this. For example when looking at a wikipage on "Malaria" observe that the section headings consist of "Introduction", "treatment" etc. Other disease related pages tend to have similar section headings. Thus the text under "Treatment" is related to the "Entity" malaria by the "treatment" relation. In other words we can define a treatment relation with two attributes "Entity" and "Treatment Text" which will contain tuples of the form "(Malaria,¡Text Description covering malaria treatment¿)".

It is clear that from the standpoint of resolving user questions, the attributes that appear frequently i.e. appear in more entities are more likely to be useful. The query templates learnt over these attributes will be more applicable in resolving questions across entities.

In order to identify the most useful attributes we rank all section headings based on the frequency of entity pages they appear in. We pick nearly 300 top headings (each appearing in atleast 25 pages) and manually analyzed them. Some irrelevant headings such as "See Also", "References" etc. were eliminated. Of the remaining, many similar headings were merged into the same attribute. For example "Signs and Symptoms", "Symptoms", "Symptoms and Signs" etc. were all grouped into the same attribute "Signs and Symptoms". Similarly "Adverse effects", "Side effects" "Side-effects" "Side Effects" "Health effects" etc. were all merged under the same attribute "Side Effects". Ultimately after pruning and merging we ended up with 59 attributes covering a total of $68K$ text values over the $29K$ entities.

The nature of these relationships is such that each text value is related to only a single entity.

Defining extended relations:

In step two we were able to define relationships between entities and various text descriptions. However the entities themselves are also related to each other. For example the drug "Chloroquine" is a "Medication" of the disease "Malaria". Incorporating such relationships into the knowledge-base is critical to resolving user questions.

A good way to identify such entity-entity relations is to simply define them based on the attribute in whose text value an entity appeared in. For example "Chloroquine" appears in the medication description of "Malaria", which is a text value related to "Malaria" via the "Medication Text" relation.

To this end we further define 59 additional relations of the form "(Entity,¡Attribute¿ Entity)" for example the relation medication entity has the attributes "(Entity,Medication Entity)" and a sample tuple "(Malaria, Chloroquine)"

As a result after defining the extended relations our final knowledge base comprised of 118 relations covering $29K$ entities and $68K$ text values. We believe there was a significant overlap between the knowledge-base and the cQA collection.

6.5.2 Mining legitimate query set (S_D)

Recall a sample sql query we discussed earlier

$s1$: `select Treatment from Rel where Symptoms = symp1`

We refer to "symp1" as a constraint and the remaining query as the template i.e.

$t1$: `select Treatment from Rel where Symptoms =`
`<some symptom value>`

Note that while query $s1$ is only useful in resolving questions that match the symptom value $symp1$, the template $t1$ allows us to generalize across questions and is useful for resolving any question that provides a symptom description and expects a treatment in response. $t1$ intuitively captures a kind of general "reasoning" that may be performed to resolve some of the questions that provide symptom descriptions. Hence we can treat any sql query that is generated by adding a constraint with template $t1$ as legitimate. More generally, our main intuition behind defining S_D is to identify a set T of such templates, and assume that any query they generate is in S_D.

Now assuming we have a training set of questions and their responses from the knowledge base available, we can use it to first discover single constraint queries, and then convert them into templates by simply dropping the constraint. More general templates will naturally be found in multiple training question-response pairs.

Thus the main challenge in mining such templates is to identify a sql query given a question, its response value and the knowledgebase. Our approach to solving this problem is to view our entire wikipedia knowledgebase as a graph. Each value in the knowledgebase (either verbose text, or entity name) is treated as a node in the graph. Any two values that appear in the same tuple are connected via an edge. Naturally any two neighboring nodes are bound to have some relationship between them. This relationship is assigned to the edge as a label.

Consider the three relations in Table 6.1 which are similar to those in our wikipedia knowledge-base. The corresponding knowledge base graph is shown in Figure 6.1.

Entity	SymptomText
$D1$	$S1$
Entity	MedicationEntity
$D1$	$M1$
$D1$	$M2$
$D2$	$M1$
Entity	AdverseEffectsText
$M1$	$A1$
$M2$	$A2$

Table 6.1: A Sample Knowledgebase to be mined

Each edge in the graph is assigned a label of the form $Attribute1\_Attribute2$ which represents the relationship between the nodes. Now assuming that our question matched the constraint $S1$ and our response contained the value $A1$, we first obtain the shortest path between the two nodes. Which in this case is $S1 \rightarrow D1 \rightarrow M1 \rightarrow A1$. Once the path is found, we traverse from the constraint node ($S1$), one step at time towards the response node ($A1$). In each step a new sql construct of the following form is added in a nested fashion.

`select` $Attribute2$ `from`
`<Relation containing` $Attribute1\_Attribute2$`> where`

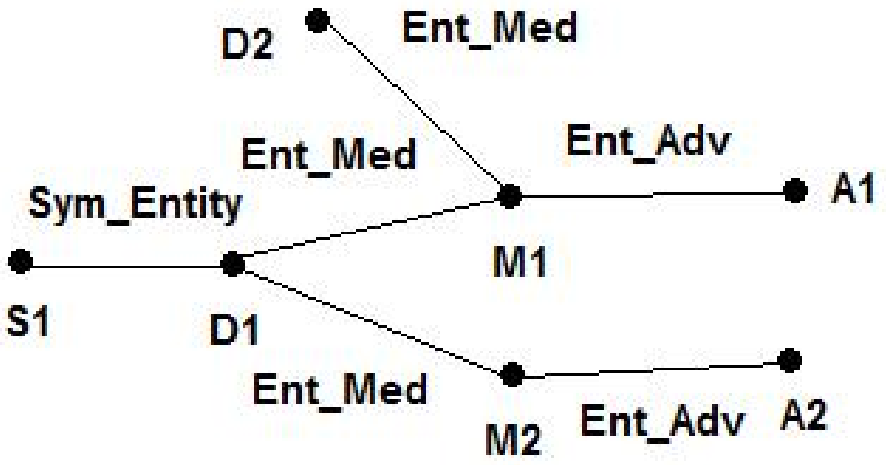

Figure 6.1: Knowledgebase in Table 6.1 viewed as a graph

$Attribute2$=<current node value>

More specifically in the first step we generate

```
select Entity from Entity_SymptomText where SymptomText = S1
```

The eventual query template generated is

```
select AdverseEffectsText from Entity_AdverseEffectsText where
Entity = (select MedicationEntity from Entity_MedicationEntity
where Entity = (select Entity from Entity_SymptomText where
SymptomText = <symptom text value>)
```

Note that the queries that get generated in this fashion may at times end up returning multiple values along with the response value. Especially because multiple entity values may result from one of the nested queries. However our model does take that into account through the value prediction model $P(v|s, q)$ and will favor queries generating fewer values.

Relation	Attributes
Disease Overview	(Disease,Overview)
Disease History	(Disease,History)
Causes	(Disease,Causes)
Medical Treatment	(Disease,Treatment)
Care at Home	(Disease,HomeCare)
Diagnosis Procedure	(Disease,Diagnosis)
Followup	(Disease,Followup)
Disease Prevention	(Disease,Prevention)
Prognosis	(Disease,Prognosis)
Disease Synonyms	(Disease,Synonyms)
Disease Symptoms	(Disease,Symptoms)
Surgery Treatments	(Disease,Surgery)
When to Seek Care	(Disease,Medical)

Table 6.2: Schema Definition for the validation database used

To summarize, for each question in the training set, we used the top 10 knowledge base values found by the constraint prediction component as our constraints i.e. start nodes, and the response value as the end node. Shortest paths were then found in each case and converted into query templates. All templates found in more than 2 questions were considered as legitimate.

6.6 Experiments

6.6.1 Validation set for training set generator

In order to validate our training set generation method and tune its parameter K, we needed to create a small fully judged collection containing questions and their responses from a knowledge base. For this we used 21 questions from Yahoo! answers cQA website and manually exhaustively labeled all their responses in a secondary database constructed from eMedicinehealth. We preferred eMedicineHealth, since it is much smaller than wikipedia making it possible to easily create a fully judged collection. The information is mainly centered around diseases and their related properties. The schema comprises 13 attributes such as $Disease$, $Treatment$, $Symptoms$ etc. The entire schema is shown in Table 6.2. In all it contained information on 2285 diseases with a total of 8825 values. The judgement set contained an average of 3.5 relevant values per question (out of a total 8825). All judgements were reviewed by a medical expert.

6.6.2 Automated evaluation

We processed $80K$ healthcare questions from Yahoo! Answers website using our automated training set generation approach to generate an automated evaluation set. Many questions were filtered out by one or more filtering criteria (see section 6.4.4). The resulting dataset contained $5.7K$ questions, for each of which a value from our wikipedia knowledge-base was identified as the response. This dataset served as a large automated evaluation set for training and evaluation of our approaches.

For automatic evaluation we used 5-fold cross validation. In each fold $4.6K$ instances were used for finding the template set S_D, learning the attribute prediction model and tuning the parameters of the constraint prediction model. The learnt model was then evaluated on the remaining $1.1K$ questions.

6.6.3 Manual evaluation

We also created a smaller manual evaluation set consisting of 60 manually judged questions. The judgements for this dataset were created by employing a pooling strategy often used in information retrieval evaluation. For each question, we pooled together top 5 values returned by different ranking methods

1. The $Sim()$ function used in constraint prediction. This represented a state of the art information retrieval method.
2. Our kbQR method trained on the automatically generated training set (excluding the 60 manually judged queries)
3. Training Set Generator

Each of these values was then judged by a medical expert as being a relevant or irrelevant response to the question. The resulting relevant values then became our final evaluation set. In all 116 relevant responses were found.

For manual evaluation, we trained on all $5.7K$ training questions excluding the 60 manually judged questions which were used for evaluation.

Evaluation metrics

We used Success at 1 ($S@1$), Success at 5 ($S@5$) and Mean Reciprocal Rank (MRR) for evaluation. Each criterion allows us to analyze a different aspect of the performance and guess utility for a specific application. $S@1$ gives us the percentage of questions resolved correctly at rank 1 and is relevant if for example we want to automatically post a response on some cQA website. On the other hand MRR and $S@5$ are more useful if we assume the user may be able to look at a short ranked list of responses.

Note that since the automatic trainingset generator is only confident about the best response, for each question in the automatic evaluation set, we only have one relevant response.

6.6.4 Experiment design

The main goal behind our experiments was to check if it was feasible to resolve questions through text mining a knowledge base and whether our kbQR approach would outperform a state of the art baseline. The baseline we primarily want to compare against is the state of the art KL-Divergence retrieval method ($Sim()$) which we use for constraint prediction. Out performing it would ascertain that the kbQR model indeed adds value over baseline retrieval. Other questions critical to success of kbQR, that we are interested in are a) To check if automated trainingset generation was feasible and b) if it was common to find query templates useful in resolving multiple questions.

6.7 Results

6.7.1 Response ranking performance

Figure 6.2 compares the 5-fold cross validation performance of our method with the constraint prediction baseline. In each fold, we use $4.6K$ questions for mining templates and estimating parameters. The rest $1.1K$ questions are used for prediction. We observe that kbQR outperforms the

state of the art KL-Divergence retrieval method by 39.44% in terms of $S@1$ and 21.17% in terms of MRR. Both improvements were found to be statistically significant using the Wilcoxon signed rank test with level $a = 0.05$. We also notice that the improvement is greater for $S@1$ than $S@5$ suggesting that kbQR is better at pushing the most relevant response to the top. Overall kbQR succeeded in resolving nearly 17% of the responses correctly. A sample response generated by kbQR in response to a question regarding cramps is shown below. It clearly shows that useful responses for cQA questions do exist in online knowledgebases and our proposed approach can help retrieve them, thereby helping the users.

Question:*How to relieve cramps in feet? I often get cramps in my feet and i have no idea as to why. Does anybody know what could cause them and any way to relieve them when they occur?*
Response by kbQR:*Skeletal muscles that cramp the most often are the calves, thighs, and arches of the foot ..... this kind of cramp is associated with strenuous activity and can be intensely painful ..... though skeletal cramps can occur while relaxing ..... It may take up to seven days for the muscle to return to a pain-free state ..... Nocturnal leg cramps are involuntary muscle contractions that occur in the calves, soles of the feet, or other muscles in the body during the night or (less commonly) while resting ..... Potential contributing factors include dehydration, low levels of certain minerals (magnesium, potassium, calcium, and sodium), and reduced blood flow through muscles attendant in prolonged sitting or lying down .... Gentle stretching and massage, putting some pressure on the affected leg by walking or standing, or taking a warm bath or shower may help to end the cramp.*

We next analyze the results on the manually judged dataset. The results are shown in Figure 6.3. We observe that the performance of kbQR is higher than the KL-Divergence method, but lower than the training set generator (TsetGen) which represents a kind of upper bound performance. The kbQR method resolves nearly 35% of the questions correctly, suggesting that the automatically generated trainingset was indeed useful for the task.

Finallly Figure 6.4 compares the automatic evaluation performance of kbQR and KL-Divergence

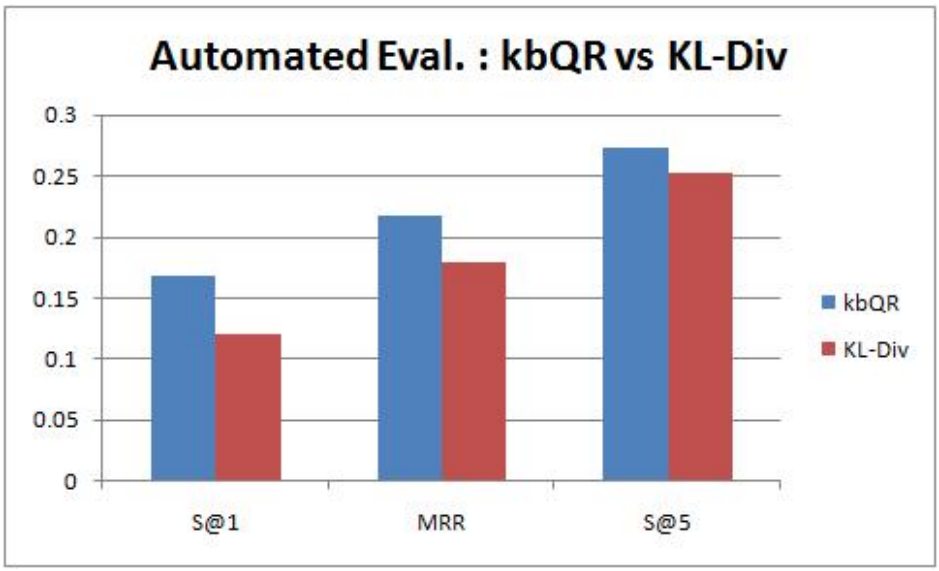

Figure 6.2: Automated evaluation performance

on the 60 questions used for manual evaluation. We assume the results of our training set generator to be the gold standard. kbQR continues to outperform the baseline. We also observe that performance is generally lower than the one obtained through manual evaluation because in the automatically generated evaluation set we are unable to generate more than one relevant result per question.

6.7.2 Training set generator validation

We validate the performance of our automated training set generator on the 21 query validation set. For each question in the validation set, we use our training set generator to find the best response from the eMedicinehealth knowledgebase and compare it to the manually judged gold standard. Note that many of the questions will get filtered out due to the filtering criteria (see section 6.4.4). The performance is evaluated only on the questions that are retained. Our goal is to find a parameter setting for K such that we retain a sufficiently high accuracy in terms of $S@1$ and MRR, without filtering out too many questions. The results are shown in Figure 6.5.

The figure shows how three different metrics $S@1$, MRR and Percentage of training questions returned vary with the parameter K. We observe that while $K = 2$ generates the most number of instances (61.9% or 13 out of 21), its accuracy is quite low. On the other hand for values 3

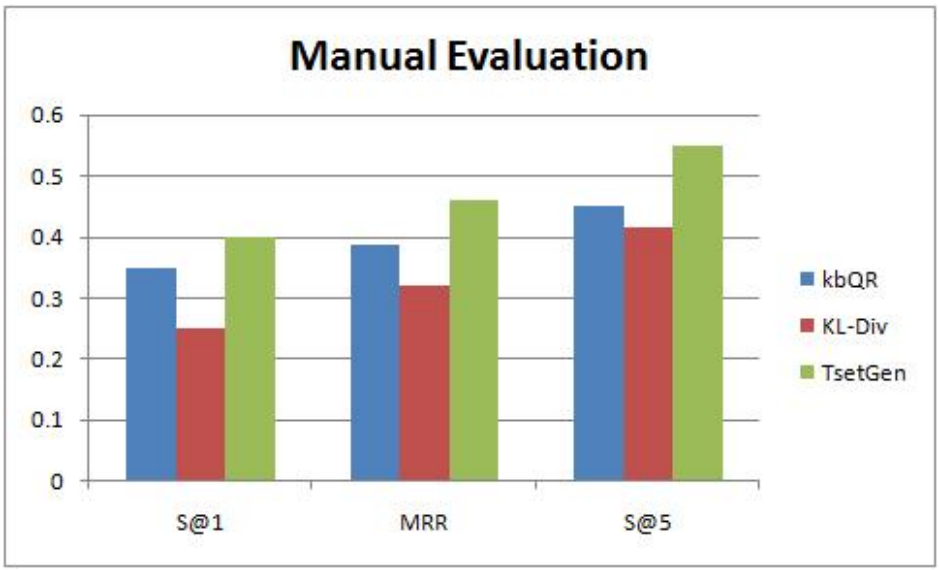

Figure 6.3: Performance on manually judged set

through 5, we retain slightly over 50% of the 21 queries as our training questions. Among these, the accuracy is highest for $K = 3$ which is the value we select for K.

6.7.3 Analysis of mined queries

Our next goal is to analyze whether it is possible to learn query templates that are reusable across questions. We executed our template mining step on all $5.7K$ training questions and plotted two graphs. The first is shown in Figure 6.6. It plots the question frequency of a query template on the x-axis and the number of query templates that were found with that frequency on the y-axis. The question frequency of a template is the number of questions this template was useful in finding responses to. For example the first data point means that there were nearly 4000 query templates that were useful in resolving 5 questions. Naturally, as we increase the question frequency, the number of templates drops. The last data point shows that there were 340 templates that were useful in resolving 20 questions. The plot clearly suggests that many templates are useful across questions.

We next need to ensure that the high frequency templates are not concentrated only among a selected few questions. To analyze this, we plot the number of questions that will become unresolvable if we used only query templates above a certain question frequency (see Figure 6.7).

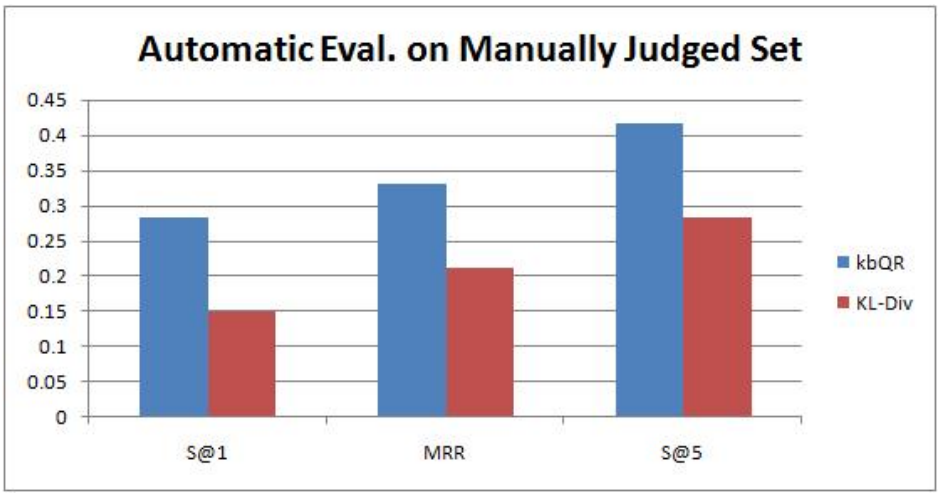

Figure 6.4: Automatic evaluation on manually judged set

For example the plot shows that if we restrict to query templates with a question frequency of 5 or more, we will be left with no templates capable of resolving 142 questions. In general we observe that for most questions there tends to be atleast 1 high question frequency template. Which is why even with threshold of 20, only 10% of the questions become unresolvable.

6.8 Discussion

In this chapter we introduced and studied a novel text mining problem, called knowledge-based question resolution, which is crucial to building autonomous agents for community question answering websites. The computational problem was to mine an online semi-structured knowledge base to discover potential responses to a natural language question on cQA sites. We proposed a general novel probabilistic framework which generates a set of relevant SQL queries and executes them to obtain responses. We presented in detail an instantiation of the general framework for resolving questions on cQA sites by leveraging the existing questions and answers as training data. Evaluation has shown that the proposed probabilistic mining approach outperforms a state of the art retrieval method and that it is indeed feasible to design autonomous agents capable of generating responses to user questions.

In the first two application tasks discussed in previous chapters, the primary challenge was to

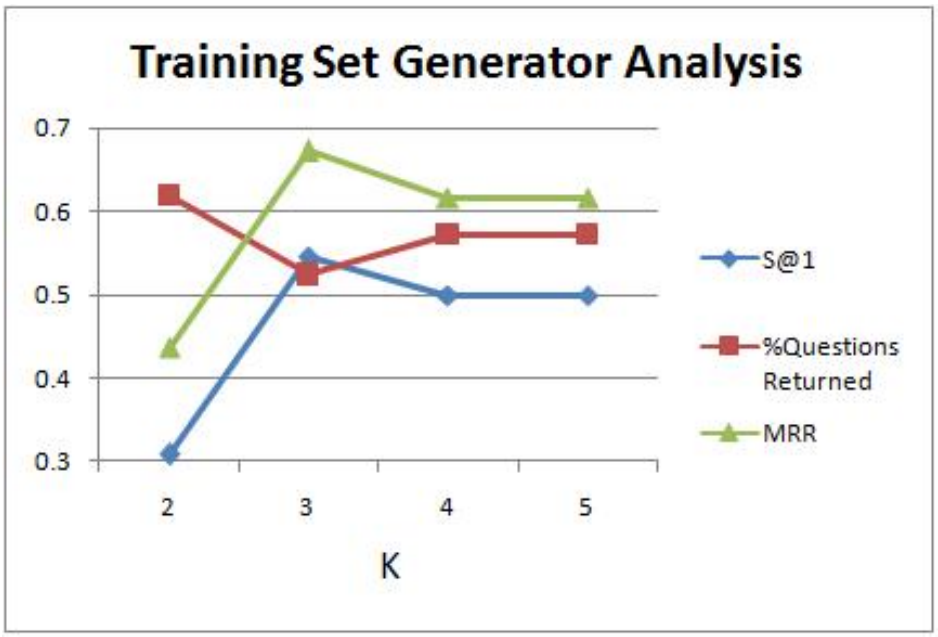

Figure 6.5: Analysis of training set generator performance on validation set($K = 3$)

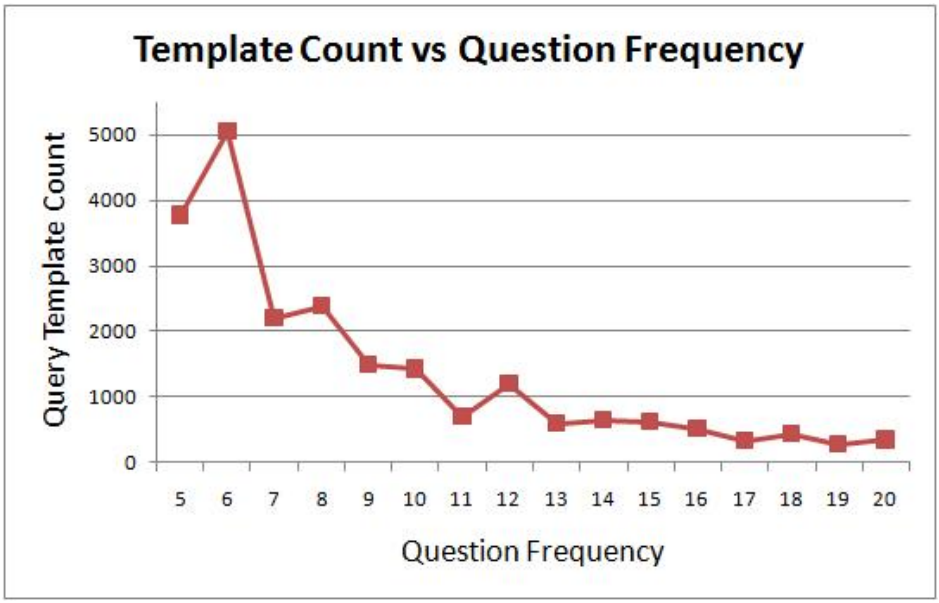

Figure 6.6: Question frequency analysis

find crucial representative keywords in a complex query. This was achieved via precise or shallow entity extraction techniques. We could then assign extracted entity keywords high weights while evaluating text similarity to achieve our final goal of finding response content similar to the query.

The task of resolving general questions however is more complicated. Here, the response is

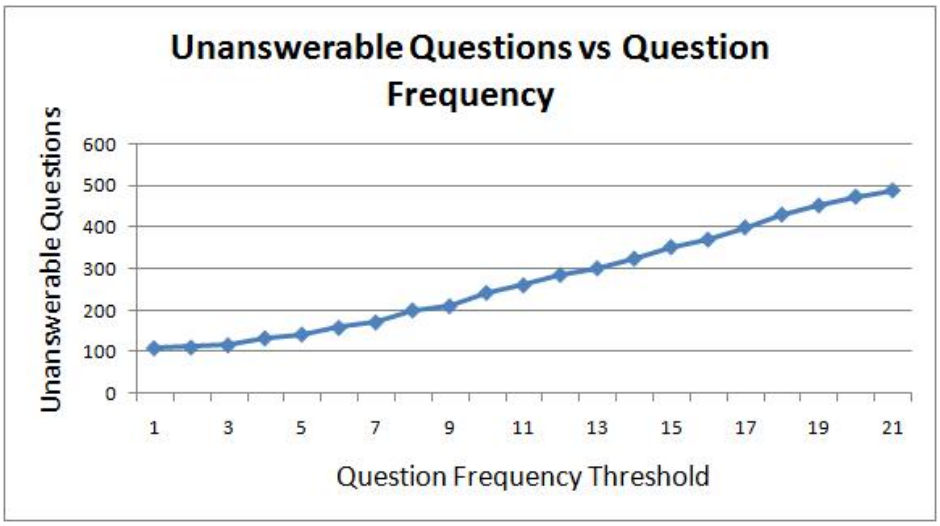

Figure 6.7: Showing the number of unresolvable questions against question frequency

not necessarily the most similar text to the question. Instead, apart from first identifying important question keywords, we also subsequently need to guess the content the user expects in response. Our proposed probabilistic framework naturally combines these two steps. The first step of estimating the constraint distribution $P(Cons(s)|q)$ in which the probability of both precise and verbose database values being present in the question is evaluated, is equivalent to precise and shallow entity extraction. In the second step relationships between values stored in the database are leveraged and appropriate sql queries are executed to generate the final response value. This second step cannot be successfully achieved unless the autonomous agent has a database of relationships between both precise and verbose text entities at its disposal. The deep semantic information necessary for the task is most easily available through health information websites, which are therefore a more suitable information source for the problem than literature or forum data.

Finally while literature data tends to be scientific and reliable, and forum data heavily anecdotal, health information websites tend to span a fairly broad spectrum in terms of reliability. From an autonomous agent's point of view, this raises another interesting challenge. Is it possible to automatically ascertain the reliability of web content being used to serve responses? We discuss our work on this question in the next chapter.

Chapter 7

Automated Reliability Prediction of Healthcare Web Content

7.1 Introduction

As the dependence on online content increases, it is important to also know what content is reliable. This is especially true in case of autonomous agents designed for the medical domain. They may be expected to provide some assurance of reliability on the content being used to generate responses.

There are some non-profit organizations, such as Health on Net Foundation (HON)[1] and Quackwatch [2], that rate websites based on how reliable they believe the website is. They do it by manually looking through a site to determine if it satisfies some conditions, such as citing references, attributing articles to experts, and so on. This task is highly effort-intensive and hence, cannot scale up well to keep pace with the rapid growth of medical information on the Web. Thus an autonomous agent seeking to index several healthcare websites, has no clue on the reliability of their content.

In this work, we want to explore if it is possible to automate this process of assessing the reliability of a webpage in the medical domain. As a first step in studying this novel problem, we focus on classifying webpages, to differentiate good informational pages from other less reliable ones. We cast the problem in a supervised learning setup and study the feasibility of learning to classify pages as reliable or not.

We propose a variety of features defined based on both the content of a webpage and other information such as links and study how different features help in this classification.

A big challenge in studying this prediction problem is that no existing test collection is available

[1] http://www.hon.ch/
[2] http://www.quackwatch.com/

Principle	Description
Authoritativeness	Qualification of the article's authors or reviewers must be present on some webpage on the website.
Complementarity	Information on the webpage should support, not replace, the doctor-patient relationship.
Privacy	Privacy and confidentiality of personal data submitted to the site by the visitor must be respected. This is required only if the page itself requires some personal information to be provided by the user.
Attribution	Source(s) of published information must be cited on some page on the site. This rule is required only if none of the authors or reviewers are qualified medical professionals.
Justifiability	Site must back up claims regarding benefits on some page on the site.
Transparency	Accessible presentation on page and email contact on some page on the site.
Financial disclosure	Funding sources must be identified on some page on the site, if the page is written by a site author.
Advertising policy	Advertising content is clearly distinguished from editorial content.

Table 7.1: Reliability criteria for medical webpages, derived from HONcode Principles used for website accreditation

for evaluation. To solve this challenge, we have created a labeled test set by leveraging the websites accredited by the Health on Net (HON) Foundation. We have also proposed appropriate measures to quantitatively evaluate this task.

Evaluation results on the dataset show that we are able to achieve an overall accuracy of over 80% in prediction. Thus, the proposed method can help significantly reduce manual labeling efforts currently in practice. Experiments also show that our prediction method works better than Google PageRank alone in reliability prediction and can be used for reranking search results based on reliability.

7.2 Notion of medical reliability

For identifying reliable pages, we define our reliability guidelines based on the eight HONcode Principles[3] (see Table 7.1). These principles are generally accepted by experts in medical community worldwide (e.g. [35]). We assume the reliability of a webpage to be a binary value (1 for reliable and 0 for unreliable), judged based on the HONcode. In reality, reliability may have multiple degrees, but similar to relevance judgments in information retrieval, assuming a binary notion of reliability makes it easier to create judgments. Further, it is not clear what principles

[3] *http://www.hon.ch/HONcode/Conduct.html*

an intermediate class ("moderately reliable") should satisfy. Manually defining criteria for such additional classes would lead to the problem of evaluating the criteria themselves. Other potential formulations, such as generating a real valued reliability score or estimating a reliability probability, also run into the same definition and evaluation hurdles. As a first step in exploring this problem, we thus restrict our study to a binary notion of reliability.

It can also be argued that even if a webpage is deemed reliable based on the HONcode principles, the content may still be inaccurate; e.g. an article based on (and citing) an inaccurate published research study. At this point, we must distinguish between *reliability* and *veracity*. Being able to extract potential facts from text and judge their veracity is not the goal of this work, but has been explored elsewhere (e.g. [86]).

7.3 Supervised learning for reliability prediction

We cast the problem of reliability prediction as a supervised binary classification problem. In a supervised setting, reliability of a webpage is defined as a binary function over computable features that model the abstract HONcode principles. We present a wide range of features in Sec. 7.3.1 and learn a Support Vector Machine classifier [85, 45] to label webpages as reliable or not.

Once the reliability of individual webpages is determined, the reliability of a website W is computed as the fraction of webpages in W found to be reliable. Thus the reliability of a website is not binary, but a real value. Our formalism fits well with the nature of the Web, where we often find a mix of reliable and unreliable pages in a website. For example, a commercial website may have some reliable pages with information about diseases, and other less reliable pages that advertise their products. Other examples include sites where both doctors and laypersons may post articles, or where some articles are properly referenced while others are not. In such cases, a binary classification of websites is insufficient to capture the diversity of the Web.

7.3.1 Features

In this section, we provide a detailed description of our proposed features. Apart from the PageRank-related feature set, all other features are calculated over individual pages.

1. Link-based Features: Links can often give a good indication on the type of webpage. For example, a reliable site is likely to contain a large number of internal links, whereas a small unreliable site is more likely to be dominated by external links of advertisements. We also defined two boolean features based on the presence of contact and privacy policy links, that are inspired by the HON reliability criteria. The absence of such information usually means the website is less reliable. The five link-based features we defined are: (a)Normalized count of internal links($\frac{\#(\text{internal links})}{Z_1}$), (b)Normalized count of external links($\frac{\#(\text{external links})}{Z_1}$), (c)Normalized count of total links($\frac{\#(\text{total links})}{Z_1}$), (d)Presence of a Contact Us Link and (e)Presence of a Privacy Policy Link. The first three are normalized count features, while the last two are binary features.

Classification models tend to perform well when all the features have nearly similar range of values. Since the number of links often vary considerably across webpages. We normalize the first three features by a sufficiently large factor Z_1. For our experiments, we set the value of $Z_1 = 200$, by observing a random sample of the dataset. (Normalizing by the maximum feature values in the dataset doesn't necessarily help as we don't know the range of values in the unseen test examples).

2. Commercial Features: Commercial interests often indicate unreliability. For example, information about a drug on a company's website may be commercially biased, and hence unreliable. To estimate if there is a commercial bias involved, we define two features based on the number of commercial keywords and commercial links:(a)Normalized count of commercial links and (b)Normalized frequency of commercial keywords in the webpage. To compute these features, we manually compiled a list of commercial words, such as *buy, sell, cheap, deal, free, guarantee, shop, price*, etc.

3. PageRank Features: PageRank provides an indication of relative "importance" of a website and has been successfully used to improve Web search performance. Moreover, unreliable sites are more likely to link to low PageRank-ed sites as compared to the reliable ones. We generated six features the first feature below represents the PageRank of the website to which the webpage belongs. The next five features are essentially a five-point representation of PageRank values of all external links [37]. We used Google PageRank(via WWW::Google::PageRank perl package) to get the PageRank values in $[0, 10]$, and we normalize it by 10 to get the values in $[0, 1]$.

(a) *Normalized internal PageRank*: $PR_{int} = \frac{PageRank(\text{parent website})}{10}$

(b) *Normalized external PageRank features* ($ExtPR$): We computed the PageRank of all websites linked from the webpage, and derived 5 features based on the five-point summary (mean, minimum, maximum, and first and third quartiles) of the values.

4. Presentation Features: Authoritative and reliable websites often seem to clearly present information, while the unreliable ones are usually cluttered with advertisements. With this idea, we define two simple presentation related features. We use elinks (*http://elinks.or.cz/*), without the frames option, to generate a text version of the webpage. Webpages cluttered with a large number of advertisements and poor presentation, when converted to text, tend to have a large number of blank lines between small scattered chunks of text. Consequently, the first feature, *Percentage of Coherent Text* ($\%CT$) is the fraction of document lines that do not have a blank line on either side. The second feature, *Percentage of Spread-out Text* ($\%ST$) is the opposite (i.e. $1 - \%CT$).

5. Word Features: The textual content and the writing style used in a webpage are usually good indicators of its reliability. For a document D, each unique word is an independent feature taking the normalized word frequency ($\frac{\#(w,D)}{\max_{w' \in D}(\#(w',D))}$) as its value.

7.4 Test set construction

Next, we wanted to build a balanced dataset that was representative of the typical webpages an Internet user might encounter. For the positive set, we used 32 medical websites that had been accredited by the HON staff during Sep–Oct 2009.[4] We applied our reliability criteria on pages from these sites and randomly selected 180 reliable pages. Since the websites had already been thoroughly reviewed and certified by experts, the task of finding reliable pages was simplified. We removed the HON seal from these pages at the time of feature generation.

For the negative set, however, we could not use this approach, since the HON website does not provide information on websites that failed the certification process. So, the negative set had to be built by directly searching for unreliable pages on the Web. We initially considered several "simple" approaches for this purpose. Intuitively, it is relatively easy to find a large number of unreliable websites by simply searching for queries like "disease name"+"what your doctor doesn't want you to know" or "disease name"+"miracle cure", etc. In addition, it is easy to find websites that promote treatments banned by the FDA [14], or the ones criticized on Quackwatch. However, it is important to ensure topical overlap between the reliable and unreliable sets of documents, so as to prevent a simple classifier from discriminating documents based solely on topic-specific keywords. Similarly, simply picking unreliable pages from obscure websites could bias the classifier to choose Page Rank as the most discriminating feature.

Therefore, for the unreliable set, we first compiled a list of topics (keywords representing diseases/conditions), covered by the 32 reliable websites. We then searched Google for (a) the topic keyword, (b) the topic keyword + "treatment", and (c) the topic keyword + "treatment" + a randomly chosen keyword from {"*cure*", "*miracle*", "*latest*", "*best*"}. For each query, we manually analyzed the webpages appearing in both the general results and advertisements, and ultimately selected 180 webpages from 35 websites that failed comprehensively on one or more of our reliability criteria. Finally, for all positive and negative pages in our dataset, we ensured that some medical information was present on the page.

[4] Information on recent certification activity is available at the "Health on Net Foundation Recent Activity" page, *http://www.hon.ch/HONcode/Patients/LatestActivity/*.

Thus, our dataset (available at *http://timan.cs.uiuc.edu/downloads.html*) consists of a total of 360 webpages divided evenly into two classes – reliable and unreliable. The size of our dataset was mainly restricted by the amount of labor needed to judge the negative documents. Since reliability analysis requires reasonable amount of expertise in understanding the criteria and the content, we chose not to use Amazon Mechanical Turk (*https://www.mturk.com/mturk/*) for data quality concerns, even though the entire process of compiling the dataset took over two weeks. Nevertheless, we believe the dataset is sufficiently large for experimenting with binary classifiers and features for reliability prediction in the sense that even with 5-fold cross validation, we still have over 72 test cases in the held-out set, which would give us a meaningful average of performance.

7.5 Experiment design

7.5.1 Evaluation measures

Our evaluation criteria are based on two prominent application settings. In the first setup, which we call as the webpage classification task, we assume that the user is surfing the Web and the classifier is required to classify every new page that the user observes. In this setting, the utility of a classifier would depend on its classification accuracy. The classifier will make two types of errors – mislabel a reliable page as unreliable (type I error) and mislabel an unreliable page as reliable (type II error). Intuitively, the type II errors would cost more. In order to account for this bias, we measure the utility of our classifiers by a weighted accuracy function, parametrized by λ:

$$\textit{Weighted Accuracy}(\lambda) = \frac{(\lambda \times TP) + TN}{\lambda \times (TP + FN) + TN + FP}$$

where unreliable pages are labeled positive, reliable pages are labeled negative, and TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively. The function assumes that cost of making a type II error is λ times the cost of making a type I error. We measure the utility of our classifiers with three different utility functions corresponding to $\lambda \in \{1, 2, 3\}$, for unbiased, moderately biased, and heavily biased setup, respectively.

In our second application setting, the system helps a human expert in labeling webpages as reliable or unreliable. We term this the webpage re-ranking task. The system generates an ordering of all webpages by ranking the reliable documents higher than all unreliable documents and the user can then look at this ordering and correct the mistakes. Ideally, the user would only need to choose a single cut-off threshold separating all reliable pages from the unreliable ones. The utility of a classifier depends on the number of mistakes that need to be corrected. This is similar to the problem of evaluating relevance ranking and, therefore, we use Mean Average Precision (MAP) as the evaluation measure for this setting.

7.5.2 Experiment procedure

For our experiments, we used the SVMlight toolkit [44] to train an SVM classifier on different feature set combinations with varying amounts of training data, for all three bias settings. For evaluation, we used 5-fold cross validation. Each fold consisted of 288 training pages (144 reliable and 144 unreliable) and 72 test pages (36 reliable and 36 unreliable). In each case, the train and test examples belonged to different sets of websites. The overall weighted accuracies and MAP scores were calculated by averaging the five values. When measuring the weighted accuracy for $\lambda \in \{2, 3\}$, the SVM classifiers were trained to account for the bias. This was realized by setting the "`-j`" parameter in SVMlight to λ. The interpretation of the parameter is the same as our interpretation of λ.

7.6 Experiment results

In this section, we first describe the results of our different lines of experiments and then present a thorough analysis of the observations. In particular, we are interested in identifying feature set combinations that lead to high performance while being robust towards amount of training data and different bias settings.

Features	Wtd. Accu. (%)			MAP		
$\lambda \Rightarrow$	1	2	3	1	2	3
Links	60.8	71.1	79.6	0.708	0.766	0.763
PageRank	72.5	77.6	**89.7**	0.856	0.846	0.866
Words	**80.6**	**83.9**	85.0	0.899	0.905	0.902
Links+Commercial	67.8	75.9	79.6	0.794	0.814	0.815
Links+Commercial+PageRank	76.4	**83.9**	86.5	0.876	0.868	0.888
All non-Word	77.2	82.4	84.6	0.873	0.863	0.881
All non-PageRank	75.8	80.6	83.5	0.886	0.890	0.893
All	80.0	83.2	86.8	**0.916**	**0.929**	**0.921**

Table 7.2: Weighted accuracy (Wtd. Accu.) and Mean Average Precision for different feature set combinations with SVM classifier

7.6.1 Effectiveness of feature sets

In our first set of experiments, we measured the performance of different feature set combinations based on overall accuracy and MAP scores. Table 7.2 shows the variation of weighted accuracy and MAP for the three bias settings over all feature sets, using SVM classifier.

Among the feature sets, word features tend to be the most discriminative, reinforcing our observation that authors of reliable and unreliable content tend to have different writing styles. PageRank features perform better than link-based features, especially when the bias is high. In such cases, we found that the internal PageRank feature, PR_{int}, becomes predominant. On the other hand, link-based classifiers use the presence of contact link CL and privacy policy link PL as dominant features. But their discriminative power is limited as many unreliable pages also contain these links and many reliable pages do not.

In general, addition of more features usually resulted in a measurable performance improvement. This is to be expected as the features belonging to different sets are largely independent and unlikely to have a high mutual information. A notable exception is the drop in performance when adding features to word based SVM classifiers. In order to better understand this behaviour, we show the MAP values of different SVM classifiers in Table 7.2. In spite of the 5% accuracy drop between *Word* and *All Non-PageRank* feature sets, the MAP value continues to remain high, suggesting that additional features are leading to a number of near misses possibly due to low performing link-based features. Similarly, while percentage accuracy of classifier based on all features is nearly same as the one trained on only word features, a higher MAP value indicates that

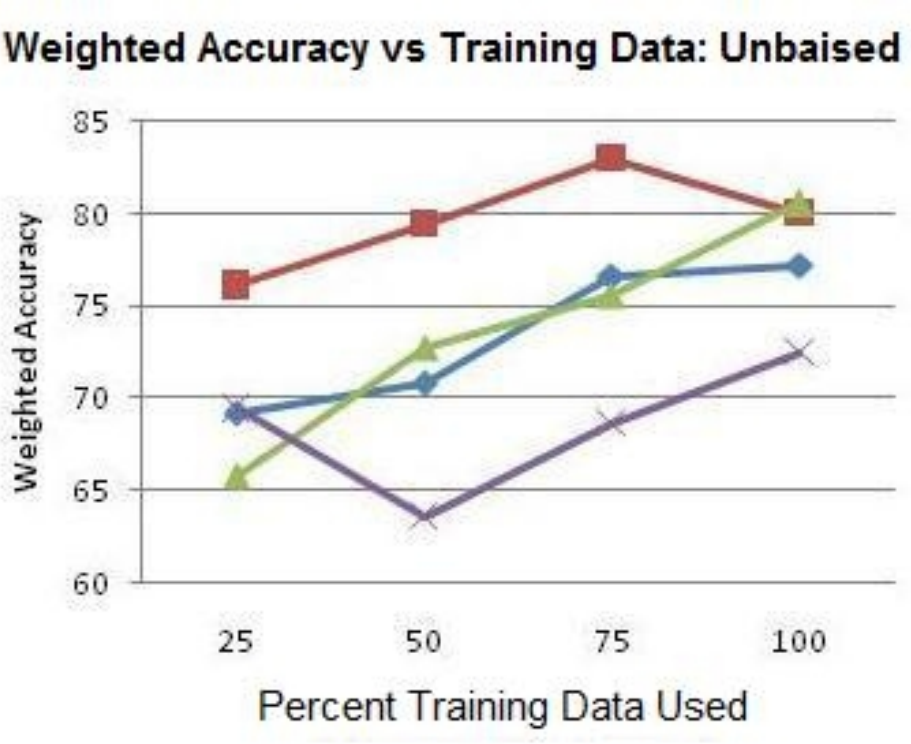

Figure 7.1: Variation of weighted accuracy with percent training data used in the unbiased case ($\lambda = 1$)

the ordering generated by the former is more accurate, making it more robust than the latter.

7.6.2 Influence of training set size

Our next line of experiments was to measure the influence of training set size on performance of different feature sets. We experimented with four high performing feature set combinations using 5-fold cross validation. For calculating performance on $x\%$ of training data, we trained each fold with only the first $x \in \{25\%, 50\%, 75\%, 100\%\}$ of training examples and tested them on the entire test set. The variation of weighted accuracy with x for $\lambda \in \{1, 2, 3\}$ are shown in Figures 7.1,7.2 and 7.3.

We observe that the classifier based on *all* features is the most robust and clearly outperforms other combinations. On the other hand, word-based features tend to perform poorly when the amount of training data is low, but their performance improves the fastest as we add more training examples. In general, both accuracy and MAP show an increasing trend with training set size, suggesting that increasing the amount of training data is likely to further improve performance. A

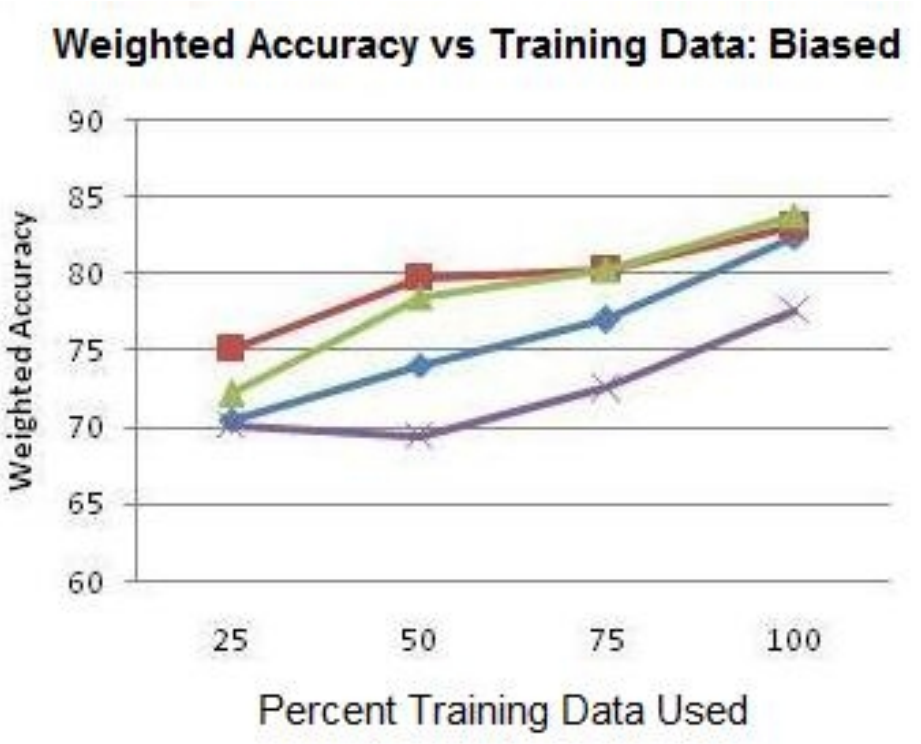

Figure 7.2: Variation of weighted accuracy with percent training data used in the moderately biased case ($\lambda = 2$)

surprising observation, however, is the fluctuation in the accuracy of PageRank based classifiers. We discuss this issue in detail below.

Issue of PageRank:

PageRank is often regarded as a crude measure of reliability. To gain a deeper insight into the performance fluctuations of PageRank features, we looked at the PageRank statistics of our dataset, shown in Fig. 7.4. The graph shows the distribution of all reliable and unreliable webpages present in the dataset based on their internal PageRank values (PR_{int}). Pages with high PageRank, in the band of $[6, 10]$, tend to be mostly reliable and, hence, easily separable. On the other hand, when the PageRank values are in $[0, 5]$, we find a mixture of reliable and unreliable pages that is hard to separate. Classifiers trained on PageRank features, tend to use $PR_{int} > \theta$ as their primary rule. Of the remaining five features, high values of $ExtPR_{min}$ (minimum $ExtPR$) and $ExtPR_{Q1}$ (first quartile of $ExtPR$) features are sometimes used for labeling pages as reliable when $PR_{int} < \theta$. The performance, therefore, mainly depends on learning an appropriate value of θ from the training

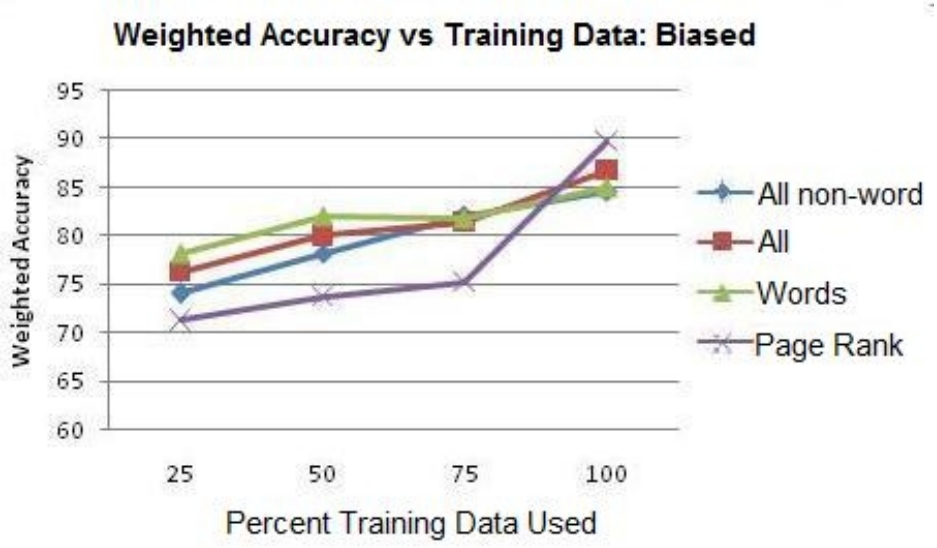

Figure 7.3: Variation of weighted accuracy with percent training data used in the heavily biased case ($\lambda = 3$)

examples. However, the narrow band between $(4, 6)$ contains a large number of both positive and negative examples. Thus, shifting θ by a single point on either side leads to high fluctuations in accuracy. For example, a simplistic classifier with only 1 rule: $PR_{int} > 4 \rightarrow Reliable$ would achieve an accuracy of 78.5% on our dataset. Raising or lowering the threshold by 1 results in a drop of 10% in accuracy. This is the reason for fluctuations in performance of PageRank classifiers. When we bias the classifier heavily, the learned classifier sets a high θ and completely disregards the remaining five PageRank features, resulting in a high reliability precision and, consequently, high weighted accuracy. We can therefore conclude that using PageRank alone as a measure of reliability is not sufficient.

7.6.3 Applications

In this section, we evaluate our classifier for two potential applications. The first is webpage re-ranking where we re-rank the results generated by a search engine based on reliability scores. The second is website accreditation, where we automatically process websites to generate a site reliability score.

Webpage Re-ranking:

For this task, we re-ranked Google's results for 22 medical queries. The queries were chosen

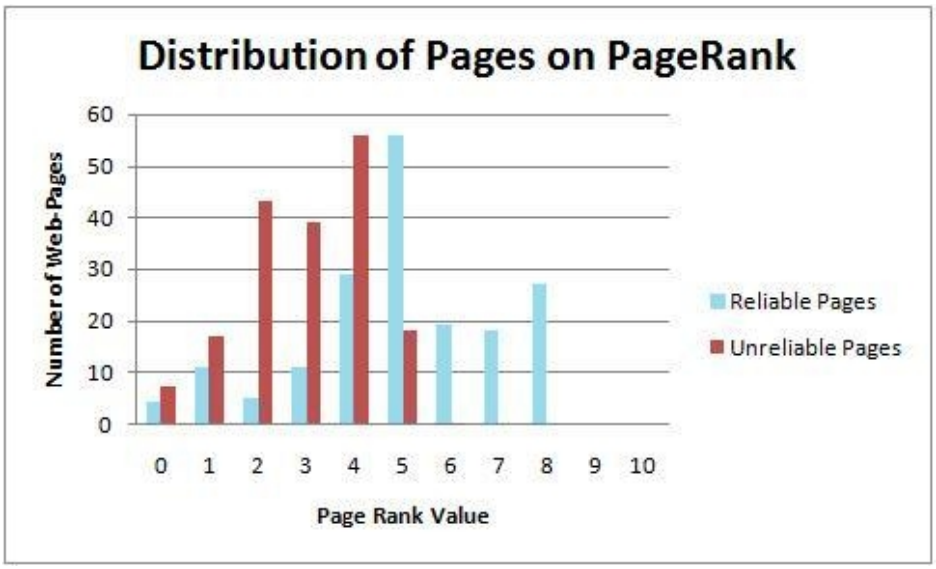

Figure 7.4: Distribution of positive and negative webpages on PageRank values

randomly from the list of "Similar Queries" displayed by Google. For each query, we manually judged the top 10 results as reliable and unreliable. We then classified each of the results using an unbiased SVM classifier ($\lambda = 1$) trained on all features. A re-ranked list was then generated based on the reliability scores. We assumed that the relevance values of all top 10 results were similar and hence our re-ranking would only slightly hurt relevance. Google's reliability MAP over 22 queries was found to be 0.753. After re-ranking, the reliability MAP improved to 0.817. The re-ranked results were found to be better in 15, worse in 5, and same in case of 2 queries. Using Wilcoxon's signed-rank test, the improvement was significant at 0.05-level. Table 7.3 shows results from Google and our re-ranking for a sample query "`cure back pain`". A main difference between the two ranked results is that the webpage *http://www.cure-back-pain.org/* was ranked the highest by Google, but our system ranked it at the bottom. When we looked at the page, we observed that it was actually a biased site which talked about the owner's own experiences and promoted a book. To summarize, these results show that even with a small training set of 360 examples, the trained classifier can already improve the quality of search results over rankings that ignore reliability. Given that our performance improves with training data, by adding more training examples, the automatic prediction method is expected to be even more useful.

Website Accreditation:

Rank	Query: `cure back pain`	
	Google	Ours
1	cure-back-pain.org	**familydoctor.org**
2	**familydoctor.org**	**emedicinehealth.com**
3	**emedicinehealth.com**	ehow.com
4	health2us.com	**webmd.com**
5	**webmd.com**	**spineuniverse.com**
6	**spineuniverse.com**	losethebackpain.com
7	ehow.com	backpaindetails.com
8	losethebackpain.com	losethebackpain.com
9	backpaindetails.com	health2us.com
10	losethebackpain.com	cure-back-pain.org
MAP	0.608	0.888

Table 7.3: Sample re-ranking results for an example query. Pages judged reliable are in bold face. Only domain names are shown for brevity

Website	Rel	Unrel
mayoclinic.com	98%	2%
rxlist.com	91%	9%
medicinenet.com	87%	13%
cancer.gov	65%	35%
goldbamboo.com	57%	43%
healthy-newage.com	51%	49%
guide4living.com	45%	55%
mnwelldir.org	43%	57%
shirleys-wellness-cafe.com	9%	91%
northstarnutritionals.com	0%	100%

Table 7.4: Websites ordered based on percentage of reliable pages found (out of 100 webpages each)

For the website accreditation task we selected a set of 10 websites and classified their webpages. None of these websites were included in our original training set. For each website, 100 webpages selected in a breadth-first manner were classified, and the percentage of reliable and unreliable pages was calculated. Classification results using a moderately biased SVM classifier ($\lambda = 2$) trained on all feature sets except PageRank are as shown in Table 7.4. The websites are ordered based on percentage of reliable pages. We observe that more authoritative and trustworthy sites, such as *www.mayoclinic.com* or *www.cancer.gov*, are ranked high. On the other hand, websites like *www.northstarnutritionals.com*, which is purely a commercial site selling online medications, and *www.shirleys-wellness-cafe.com*, which is an alternative medicine website not conforming to most of the HON criteria and containing content strongly critical of modern medicine, are ranked lowest. Websites like *www.guide4living.com* and *www.healthy-newage.com*, which are not particularly authoritative, conform to only some of HON criteria and provide mostly unbiased non-commercial information are ranked in the middle. Thus, our system generated a reasonable overall ranking of websites. We did not use the PageRank features for these experiments, as PageRank values need to be requested from an external Google Web Service that does not serve the requisite high volume of requests generated for obtaining external PageRank features, $ExtPR$. Additional website accreditation experiments with upto 5000 pages per website returned similar results.

7.7 Discussion

In this chapter, we presented a study of automatically predicting reliability of webpages in the medical domain. This can be useful for identifying reliable online resources from which an autonomous agent may extract content to generate responses. We cast the problem in a supervised learning setup and also created a publicly available test set to quantitatively evaluate the task. Experimental results on this dataset are very encouraging. We were able to achieve an overall accuracy of 80%, showing that it is indeed feasible to predict the reliability of medical webpages through automatic feature extraction and classification. Results further show that using all the types of proposed features works better than only some of them, and performance can generally be improved over the Google PageRank baseline. Due to the importance of reliability in medical domain, we believe that our study can potentially have an impact on helping users to better assess reliability of information on the Web in this very important domain.

Chapter 8

Conclusions and Directions for Future Work

In this thesis we introduced a novel paradigm for information service in which an autonomous agent proactively helps users on web communities by posting responses to their unresolved questions. Such a paradigm has significant advantages over traditional search and recommendation systems in helping resolve users' complex information needs. The major contribution of this thesis was to show that it is indeed feasible to build agents capable of generating meaningful responses to user queries with a high accuracy in the healthcare domain, and this claim is supported in three case studies.

The first study involved designing an agent for resolving similar case-based queries using literature data, targeted mainly at physician users. This problem was close to a traditional retrieval problem, except for an important distinction; the queries involved were long and complex. We addressed the problem by designing novel methods that utilized biomedical semantic resources for identifying keywords representing critical entities and showed that a precision at top 10 documents of upto 0.48 could be achieved, meaning that on average nearly 50% of the cases in the system-generated response would be useful to the user. Thus it is indeed feasible to build autonomous agents for this application.

The second study involved resolving similar case-based queries using web forum data. This task was intended to serve users of web forums, who are predominantly laypersons and cannot understand medical literature. The task was more challenging in that the queries tended to have noisy non-technical language and a lot of case unrelated background information. Also the semantic resources found to be so helpful in the first task, were no longer suitable for forum data. We instead based our methods on shallow semantic information extraction techniques that separated

case related sentences from the background. The best performing method achieved a precision at top 5 documents of 0.54 highlighting the feasibility of automatic response generation for this application.

Finally the third study required building an agent for resolving general healthcare questions found in community question answering websites. Generic questions are harder to resolve compared to similar case-based queries that were dealt with in the first two tasks. This task required more detailed semantic information in the form of a database containing precise medical entities, verbose text descriptions, and the relations between entities and text descriptions. Health information websites were found to be the most useful resource for building such a database and hence were better suited as an information source than web forums or medical literature. We proposed a principled probabilistic framework for the problem that utilized the entity-relation database for generating responses. Our proposed approach was able to resolve over 30% of the questions posed in our data set of 60 manually judged healthcare questions obtained from the Yahoo! Answers website. suggesting the task was indeed feasible.

The envisioned new paradigm of proactive information service logically contains four high level technical challenges:

1. Detection of an unresolved information need i.e. query/question
2. Identification of relevant information from the information source
3. Generation and posting of the response
4. Gathering user feedback to evaluate end user utility

Out of these four, the second challenge of being able to find relevant information to a query to generate a response is the most important in ascertaining the feasibility of the paradigm. Thus our focus in this thesis has been on addressing the *relevance* challenge. For the remaining three challenges, we have either used or it is possible to use intelligent baselines. For example, we can detect all queries with no responses as unresolved information needs. In response generation, we have used pre-defined response formats. For evaluation, we have used manual judgments provided

by third party experts as surrogates for feedback from actual users who posted those queries online. Now that the feasibility has been ascertained, in future we plan to study a number of new research questions that arise from the remaining three challenges. We discuss some of them below.

8.1 When should we consider an information need unresolved?

An important aspect for future study is understanding when an information need is unresolved. Some web communities specifically allow users to mark queries as *Resolved* once a satisfactory response is provided. But this is not always the case. A trivial alternate is to just check if no responses were provided by other users. In general however the problem of identifying unresolved queries is non-trivial and one needs to analyze all of the human user posted responses before making a decision. Thus it requires further study.

8.2 What is the best response format?

Another important problem is to understand the nature of responses users prefer the most, and how we can generate them. This raises several important questions.

- Do users only care about the relevance of information, or do they also care about how 'human like' a response is? For example, users may be more likely to accept a response if it contained comforting sentences such as *I'm sorry you're suffering so much* or *Hope you feel better* etc. And if this is indeed true, how can we generate more human like answers?
- What is an appropriate size of response? This affects the number of sentences we can extract from a relevant document to include in the response. For example, in our second application task, should the response also contain text snippets from each similar discussion thread? More importantly, how do we select these relevant sentences from one or more relevant documents?
- Finally it is also important to learn how the answers to above questions vary depending upon the web community, domain, user base or information source being targeted.

8.3 When should a response be posted?

It is also important to understand the amount of time the agent should wait before posting a response. Here we need to make a trade-off between resolving the user's information need on one hand, and not interfering unnecessarily in the normal human interactions on the other. Is it better to respond as soon as a query is posted, or should the agent allow some time for other users to post their responses? One option is to always provide a fixed time lag (Eg. one day). However a more useful approach would be one where the system detects the urgency in the question and dynamically adjusts its time lag.

8.4 How to evaluate the end user utility?

Our proposed methods in all three tasks were evaluated by measuring their accuracy on manually judged evaluation sets. While such an evaluation is well suited for comparing methods and ascertaining feasibility, it does not provide a complete picture of the utility the agent will provide to end users. This is because the relevance judgments were provided by human experts other than the end users who asked the queries.

Ideally the only way to evaluate end user utility of an agent is by posting responses to queries on the web community, and then analyzing feedback from users who posted the query. However, in practice user feedback tends to be quite sparse. Web communities like forums often don't have any mechanism for quantitative feedback. Community question answering services like Yahoo! Answers do allow users to rate/like responses, but these features are frequently ignored. Overall a vast number of posted responses don't receive any quantitative/qualitative feedback from users making it difficult to ascertain whether the user liked/disliked them.

References

[1] The gnu image-finding tool. http://www.gnu.org/software/gift/.

[2] Imageclef medical case retrieval task. http://www.imageclef.org/2010/medical.

[3] Medical subject headings. http://www.nlm.nih.gov/mesh/.

[4] The metamap transfer toolkit. http://mmtx.nlm.nih.gov/.

[5] Pubmed clinical queries. http://www.ncbi.nlm.nih.gov/pubmed/clinical.

[6] Pubmed search engine. http://www.ncbi.nlm.nih.gov/pubmed.

[7] Trec genomics track. http://ir.ohsu.edu/genomics/.

[8] Trec question answering track. http://trec.nist.gov/data/qamain.html.

[9] The unified medical language system. http://www.nlm.nih.gov/research/umls/.

[10] *Medical image retrieval based on visual contents and text information*, volume 1, 2004.

[11] A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1):39–59, Mar. 1994.

[12] E. Agichtein, Y. Liu, and J. Bian. Modeling information-seeker satisfaction in community question answering. *ACM Trans. Knowl. Discov. Data*, 3(2):10:1–10:27, Apr. 2009.

[13] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, K. Jain, V. Mirrokni, and S. Teng. Robust PageRank and Locally Computable Spam Detection Features. In *AIRWeb '08: Proceedings of the 4th Intl. Workshop on Adversarial Information Retrieval on the Web*, pages 69–76, 2008.

[14] Y. Aphinyanaphongs and C. F. Aliferis. Text Categorization Models for Identifying Unproven Cancer Treatments on the Web. In *MedInfo*, pages 968–972, 2007.

[15] S. J. Athenikos and H. Han. Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1–24, 2010.

[16] T. Baldwin, D. Martinez, and R. B. Penman. Automatic thread classification for linux user forum information access. 2008.

[17] L. Becchetti, C. Castillo, D. Donato, R. Baeza-Yates, and S. Leonardi. Link analysis for Web spam detection. *ACM Trans. Web*, 2(1):1–42, 2008.

[18] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Inf. Process. Manage.*, 43(4):866–886, July 2007.

[19] I. Bichindaritz and C. Marling. Case-based reasoning in the health sciences: What's next? *Artif. Intell. Med.*, 36(2):127–135, Feb. 2006.

[20] M. W. Bilotti, J. Elsas, J. Carbonell, and E. Nyberg. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 459–468, New York, NY, USA, 2010. ACM.

[21] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link Analysis Ranking: Algorithms, Theory, and Experiments. *ACM TOIT*, 5(1):231–297, 2005.

[22] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

[23] M. Braschler, D. Harman, and E. Pianta, editors. *CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy*, 2010.

[24] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proc. of WWW*, 1998.

[25] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Math. Program.*, 63(2):129–156, Jan. 1994.

[26] A. B. Can and N. Baykal. MedicoPort: A medical search engine for all. *Comput. Methods Prog. Biomed.*, 86(1):73–86, Apr. 2007.

[27] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474, New York, NY, USA, 2008. ACM.

[28] H. Cui, M.-Y. Kan, and T.-S. Chua. Generic soft pattern models for definitional question answering. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 384–391, New York, NY, USA, 2005. ACM.

[29] D. Dinh and L. Tamine. Irit at imageclef 2010: Medical retrieval track. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

[30] H. Duan and C. Zhai. Exploiting thread structures to improve smoothing of language models for forum post retrieval. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 350–361, Berlin, Heidelberg, 2011. Springer-Verlag.

[31] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 347–354, New York, NY, USA, 2008. ACM.

[32] J. L. Elsas and J. G. Carbonell. It pays to be picky: an evaluation of thread retrieval in online forums. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 714–715, New York, NY, USA, 2009. ACM.

[33] J. Erinjeri and S. Bhalla. Redefining radiology education for first-year medical students: shifting from a passive to an active case-based approach. *Acad Radiol*, 13(6):789–96, 2006.

[34] A. Gaudinat, N. Grabar, and C. Boyer. Automatic Retrieval of Web Pages with Standards of Ethics and Trustworthiness Within a Medical Portal: What a Page Name Tells Us. In *Proc. of Conf. on Artificial Intelligence in Medicine (AIME)*, pages 185–189, 2007.

[35] A. Gaudinat, N. Grabar, and C. Boyer. Machine Learning Approach for Automatic Quality Criteria Detection of Health Web Pages. In *Proc. of the World Congress on Health (Medical) Informatics – Building Sustainable Health Systems*, volume 129, pages 705–709, 2007.

[36] G. H. Guyatt, M. O. Meade, R. Z. Jaeschke, D. J. Cook, and R. B. Haynes. Practitioners of evidence based care. *BMJ*, 320(7240):954–955, 4 2000.

[37] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publ., 2006.

[38] M. R. Henzinger. Link Analysis in Web Information Retrieval. In *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, volume 23, pages 3–8, 2000.

[39] W. R. Hersh, C. Buckley, T. J. Leone, and D. H. Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 192–201. ACM/Springer, 1994.

[40] X. Huang and Q. Hu. A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA, 2009. ACM.

[41] Z. Huang, M. Thint, and A. Celikyilmaz. Investigation of question classifier in question answering. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 543–550, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[42] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 84–90, New York, NY, USA, 2005. ACM.

[43] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 76–83, New York, NY, USA, 2005. ACM.

[44] T. Joachims. Making large-scale SVM Learning Practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.

[45] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of ECML*, pages 137–142, 1998.

[46] D. B. Johnson, W. W. Chu, J. D. Dionisio, R. K. Taira, and H. Kangarloo. Creating and indexing teaching files from free-text patient reports. *Proc AMIA Symp*, pages 814–8, 1999.

[47] K. . Jones and C. J. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical Report British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

[48] J. M. Juarez, J. Salort, J. Palma, and R. Marin. Case representation ontology for case retrieval systems in medical domains. In *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, AIAP'07, pages 168–173, Anaheim, CA, USA, 2007. ACTA Press.

[49] J. Kim, X. Xue, and W. B. Croft. A probabilistic retrieval model for semistructured data. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 228–239, Berlin, Heidelberg, 2009. Springer-Verlag.

[50] J. Y. Kim and W. B. Croft. A field relevance model for structured document retrieval. In *Proceedings of the 34th European conference on Advances in Information Retrieval*, ECIR'12, pages 97–108, Berlin, Heidelberg, 2012. Springer-Verlag.

[51] D. R. Lankes. *Trusting the Internet: New Approaches to Credibility Tools*, pages 101–122. MIT Press, 2008.

[52] Q. Liu, E. Agichtein, G. Dror, Y. Maarek, and I. Szpektor. When web search fails, searchers become askers: understanding the transition. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 801–810, New York, NY, USA, 2012. ACM.

[53] Z. Lu. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011, Jan. 2011.

[54] Z. Lu, W. Kim, and W. J. Wilbur. Viewpoint paper: Evaluating relevance ranking strategies for medline retrieval. *JAMIA*, 16(1):32–36, 2009.

[55] D. L. Maclean and J. Heer. Identifying medical terms in patient-authored text: a crowdsourcing-based approach. *J Am Med Inform Assoc*, May 2013.

[56] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[57] J. V. Marriott, P. Stec, T. El-Toukhy, Y. Khalaf, P. Braude, and A. Coomarasamy. Infertility information on the World Wide Web: a cross-sectional survey of quality of infertility information on the internet in the UK. In *Human Reproduction*, pages 1520–1525, Jul 2008.

[58] M. J. Martin. *Reliability and verification of natural language text on the world wide web*. PhD thesis, Las Cruces, NM, USA, 2005. Chair-Hartley, Roger T.

[59] S. C. Matthews, A. Camacho, P. J. Mills, and J. E. Dimsdale. The Internet for Medical Information About Cancer: Help or Hindrance? In *Psychosomatics*, volume 44, pages 100–103, Apr 2003.

[60] M. Minock. C-phrase: A system for building robust natural language interfaces to databases. *Data Knowl. Eng.*, 69(3):290–302, Mar. 2010.

[61] H. Müller, J. Kalpathy-Cramer, I. Eggel, S. Bedrick, S. Radhouani, B. Bakke, C. E. K. Jr., and W. R. Hersh. Overview of the clef 2009 medical image retrieval track. In *CLEF (2)*, pages 72–84, 2009.

[62] K. Pattabiraman, P. Sondhi, and C. Zhai. Exploiting forum thread structures to improve thread clustering. ICTIR '13, 2013.

[63] V. Petras, P. Forner, and P. D. Clough, editors. *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, 2011.

[64] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[65] A.-M. Popescu, A. Armanasu, O. Etzioni, D. Ko, and A. Yates. Modern natural language interfaces to databases: composing statistical parsing with semantic tractability. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[66] S. L. Price and W. R. Hersh. Filtering Web pages for Quality Indicators: An Empirical Approach to Finding High Quality Consumer Health Information on the World Wide Web. In *Proceedings of AMIA Symposium*, pages 911–915, 1999.

[67] G. Quellec, M. Lamard, L. Bekri, G. Cazuguel, C. Roux, and B. Cochener. Medical case retrieval from a committee of decision trees. *Trans. Info. Tech. Biomed.*, 14(5):1227–1235, Sept. 2010.

[68] G. D. Rennels, E. H. Shortliffe, F. E. Stockdale, and P. L. Miller. A computational model of reasoning from the clinical literature. *AI Magazine*, 10(1):49–57, 1989.

[69] S. Robertson, S. Walker, M. Beaulieu, and P. Willett. Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track. *In*, 21:253–264, 1999.

[70] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM.

[71] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track.

[72] A. Rosset, H. Müller, M. Martins, N. Dfouni, J.-P. Vallée, and O. Ratib. Casimage project - a digital teaching files authoring environment. *Journal of Thoracic Imaging*, 19(2):1–6, 2004.

[73] V. L. Rubin and E. D. Liddy. Assessing credibility of weblogs. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 187–190, 2006.

[74] M. E. Ruiz. Combining image features, case descriptions and umls concepts to improve retrieval of medical images. *AMIA Annu Symp Proc*, pages 674–8, 2006.

[75] D. L. Sackett, S. E. Straus MD, S. R. MD, W. Rosenberg, and B. H. MD. *Evidence-Based Medicine: How to Practice and Teach EBM (Book with CD-ROM)*. Churchill Livingstone, 2nd edition, Feb. 2000.

[76] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[77] J. Seo, W. B. Croft, and D. A. Smith. Online community search using thread structure. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1907–1910, New York, NY, USA, 2009. ACM.

[78] C. Shah. Measuring effectiveness and user satisfaction in yahoo! answers. *First Monday*, 16(2), 2011.

[79] M. Siadaty, J. Shu, and W. Knaus. Relemed: sentence-level search engine with relevance score for the MEDLINE database of biomedical articles. *BMC Medical Informatics and Decision Making*, 7(1):1+, Jan. 2007.

[80] A. Singh, D. P, and D. Raghu. Retrieving similar discussion forum threads: a structure based approach. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 135–144, New York, NY, USA, 2012. ACM.

[81] R. Soricut and E. Brill. Automatic question answering using the web: Beyond the factoid. *Inf. Retr.*, 9(2):191–206, Mar. 2006.

[82] D. J. States, A. S. Ade, Z. C. Wright, A. V. Bookvich, and B. D. Athey. MiSearch adaptive pubMed search tool. *Bioinformatics*, 25(7):974–976, Apr. 2009.

[83] V. Tablan, D. Damljanovic, and K. Bontcheva. A natural language query interface to structured information. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ESWC'08, pages 361–375, Berlin, Heidelberg, 2008. Springer-Verlag.

[84] T. T. Tang, N. Craswell, D. Hawking, K. Griffiths, and H. Christensen. Quality and relevance of domain-specific search: A case study in mental health. *Inf. Retr.*, 9(2):207–225, 2006.

[85] V. N. Vapnik. The Nature of Statistical Learning Theory. In *Springer*, 1995.

[86] V. Vydiswaran, C. Zhai, and D. Roth. Content-driven Trust Propagation Framework. In *Proceedings of the* 17^{th} *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 974–982, 2011.

[87] Y. Wang and R. Richard. Rule-based Automatic Criteria Detection for Assessing Quality of Online Health Information. *Journal on Information Technology in Healthcare*, 5(5):288–299, 2007.

[88] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[89] N. L. Wilczynski, R. B. Haynes, J. N. Lavis, R. Ramkissoonsingh, and A. E. Arnold-Oatley. Optimal search strategies for detecting health services research studies in MEDLINE. *CMAJ*, 171(10):1179–85+, 2004.

[90] P. S. Worral, C. Randolph, and R. F. Levin. Bringing evidence to the point of care. *Res Theory Nurs Pract*, 22(4):225–7, 2008.

[91] H. Wu, C. Hu, and S. Chen. Uestc at imageclef 2010 medical retrieval task, 2010.

[92] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM.

[93] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 475–482, New York, NY, USA, 2008. ACM.

[94] H. Yu, T. Kim, J. Oh, I. Ko, S. Kim, and W.-S. Han. Enabling multi-level relevance feedback on pubmed by integrating rank learning into dbms. In *Proceedings of the third international workshop on Data and text mining in bioinformatics*, DTMBIO '09, pages 43–50, New York, NY, USA, 2009. ACM.

[95] C. Zhai. *Statistical Language Models for Information Retrieval*. Now Publishers Inc., Hanover, MA, USA, 2008.

[96] C. Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, Mar. 2008.

[97] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 403–410, New York, NY, USA, 2001. ACM.

[98] L. Zhang, Y. Zhang, Y. Zhang, and X. Li. Exploring both Content and Link Quality for Anti-Spamming. In *Proceedings of the Sixth IEEE International Conference on Computer and Information Technology (CIT)*, page 37, 2006.